FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Motorcycle HUD for navigation, communication and performance monitoring

Diogo Sanches Mendes

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Dirk Elias (Prof. Dr.)

July 29th 2015

Motorcycle HUD for navigation, communication and performance monitoring

Diogo Sanches Mendes

Mestrado Integrado em Engenharia Informática e Computação

Approved by the Jury:

President:

External member:

Supervisor: Dirk Elias (Prof. Dr.)

____

July 29th 2015

Resumo

Recentemente, com os grandes avanços tecnológicos feitos regularmente na área dos meios de transporte privados, existem ainda muitas hipóteses para estudar e experiências a realizar. O objetivo é maximizar a experiência, segurança e produtividade do condutor, e ao mesmo tempo minimizar os seus riscos, bem como outros tipos de perda (consumo de combustível, tempo, ...). Estes avanços tendem a focar-se mais em carros em comparação com motociclos, em parte devido ao facto de os carros terem mais espaço e condições disponíveis para a implementação de novos tipos de interação humano-máquina.

Uma solução que tem atraído cada vez mais atenção é a utilização de Head-up Display (HUD), um ecrã ou projeção que (no caso da condução) disponibiliza vários tipos de informação aos condutores sem requerer que estes desviem a sua atenção do seu ponto de foco usual, nomeadamente a estrada.

Com a difusão dos smartphones, aplicações mobile e smartwear, surgem novas oportunidades para o desenvolvimento de HUDs para motociclos.

Esta dissertação pretende desenvolver um HUD nesses moldes, utilizando o Android OS para um smartwear específico, nomeadamente o dispositivo de hardware “Snow2”, pertencente à empresa Recon Instruments, bem como para o smartphone como aplicação de suporte. Os seus objetivos consistem em disponibilizar ao condutor de motociclo informação referente a navegação, comunicação e monitorização de desempenho, entre outro tipo de informações potencialmente interessantes, com vista a melhorar a experiência da condução de uma forma inovadora, não só em termos de utilidade, mas também de segurança e prazer na condução.

Abstract

Recently, with the great technological advances being made on a regular basis regarding personal vehicular transportation, there are yet many hypotheses to study and experiences to conduct. The main goal consists of maximizing the driver's experience, safety and productivity while minimizing his/her risks and other types of losses (fuel consumption, time, ...). These advances tend to focus more on cars in comparison to powered two-wheelers (PTW), in part due to the fact that the former offers more space and conditions for the implementation of new types of human machine interactions (HMI).

One solution that has been attracting increasing attention is the use of a Head-up Display (HUD), a screen or projection that (in the case of driving) displays various types of information relevant to drivers without requiring them to divert their attention from their usual view or focus.

With the propagation of smartphones, mobile applications and mobile wear, new opportunities for the development of HUDs for PTW drivers arise.

This dissertation aims to develop such a HUD using the Android OS for a specific smart wear, namely the "SNOW 2" hardware device developed by Recon Instruments, as well as for the smartphone as a support application. Its main goals are to provide the PTW driver with information regarding navigation, communication and performance monitoring, among other potentially interesting information, in order to enhance the driving experience in a completely new way, not only in terms of utility, but also safety and enjoyability.

"It had long since come to my attention that people of accomplishment rarely sat back and let things happen to them. They went out and happened to things."

Leonardo da Vinci

Content

List of Figures

List of Tables

Abbreviations and Symbols

HUD	Head-up Display
HDD	Head-down Display
PTW	Powered Two Wheelers
API	Application Programming Interface
HMI	Human Machine Interface
OS	Operating System
BLE	Bluetooth Low Energy
IDE	Integrated Development Environment
AS	Android Studio
SDK	Software Development Kit
OBD	On-board diagnostics

Chapter 1

Introduction

1.1 Context

In the world of today, smartphones, although being a relatively recent technology, are now massively diffused. Smartwear follows this trend and is now quite widespread, be it in the form of watches, bracelets, glasses or other types of devices. This has lead to a great focus on mobile application development, which is growing annualy at a rate of 27% (Mobile 2014).

Simultaneously, Advanced Driver Assistance Systems (ADAS) are practically common in-vehicle components. However, these advances are much more common in automobiles. Nevertheless, PTW drivers are much (26 times) more likely to be in a fatal accident as car occupants, making them one of the most vulnerable road users. In fact, PTW fatalities account for almost 20% of all fatalities in road accidents in Europe (Pieve, Tesauri, and Spadoni 2009, 603).

The Head-up Display solution for vehicle assistance has been the subject of several studies in the last years. It has been proven to improve not only the information conveyance, but also driver safety (Jun, Hang, and Wenxia 2014, 484), (Ablassmeier et al. 2007). Advantages of HUDs include quicker recongnition of displayed information, a smaller amount of eye accommodation, a combination of information between the HUD's display and the forward view of the road, and of course not having to divert user attention from the road (Todoriki et al. 1994).

This dissertation aims to mitigate the described problems by developing a driving assistance system for PTW drivers, implemented in an HUD, in this way taking advantage of the aforementioned advantages it implies. Since its results will be utilised for a project being developed in Fraunhofer Portugal, for the Fraunhofer Portugal Research Center for Assistive Information and Communication Solutions (Fraunhofer AICOS), a scholarship has been offered.

1.2 Motivation and Objectives

The motivation for the development of this project consists of developing an HUD system to assist PTW drivers in their driving. Drivers, in this context, include every-day drivers that use PTWs in their routine, long-distance drivers that travel on their motorcycles, and even amateur or professional racers.

The three main goals of this system are ease of use, affordability and utility. It should be as simple and intuitive as possible, effectively conveying all useful information to the driver while being in an acceptable price range, by combining off-the-shelf components (Snow2 HUD (please refer to Section 3.3), Android application for the HUD and connection with driver's smartphone). These goals should be achieved in such a way that driver security and safety is not only assured, but also even improved.

Although some motorcycle HUDs already exist or are currently in development (please refer to Section 2.2.2), none of them combines all three characteristics mentioned above. Thus, the motivation for this project's development arose.

In regard to specific functionalities, the basic information to be displayed is instantaneous speed and time. Other optional details to be shown are weather related, namely humidity, wind speed or temperature. One of the main features is the navigation system, which guides the driver through GPS with both visual and audio inputs. The HUD's GPS sensor provides constant updates on the user's location, and through the use of Google's Directions API, directions are fetched whenever necessary. Another aspect of the system is tour tracking. This is also achieved through GPS, and may allow not only the possibility to repeat or check completed tours, but also to analyse some performance aspects like e.g. time/distance or time/tour ratios for optimal route selection. Furthermore, drive performance will also be logged and analysed, namely average and maximum speed, total covered distance, and total driving time, among other information. Another important functionality that has an important impact on driver security is call monitoring, both incoming and outgoing. Taking into account that Snow2 does not include a microphone or speakers, a third-party Bluetooth headset is utilised to allow bilateral communication. Additionally, SnowOS natively handles incoming call and text message notifications. Text message sending is not supported for security and practicality reasons. Both calls and text messages obviously require communication with the driver's smartphone. Other secondary features that will soon be implemented are, among others, an audio player and an automatic emergency call system in case of crash detection, indicating driver name, age, blood type, and other relevant medical data.

Finally, one of the system's most important characteristics will be the layout design. Having driver security as a core objective, the interface must be simple, intuitive, and above all non-distractive, in order to convey all the necessary information without jeopardizing the user's safety. For more details concerning this matter, please refer to Chapter 5.

1.3 Dissertation structure

This document contains 7 more chapters besides the first introductory one. Chapter 2 deals with the literature review and state of the art, detailing current HUD solutions and alternatives to HUDs. Chapter 3 gives an overview of the project's software and technology specifications, including used IDE, APIs and SDKs, as well as detailed description of the HUD that will be used for the development of the project and how it was mounted on a motorcycle helmet to create the project's first functional prototype. Chapter 4 details the general system architecture and the individual modules that make up the system, also including the different ways of communication between the devices and the communication protocol. In Chapter 5, all details in regard to the layout design can be found. The 6th Chapter deals with the methodology used for the development and validation of the project. Chapter 7 contains all information regarding the conducted experiment, as well as a planned one that is ready to be carried out. Finally, Chapter 8 encompasses the conclusions of the work done so far and includes a set of functionalities to be implemented during future work.

Chapter 2

Literature Review

This chapter analyses relevant papers and other scientific documents that deal with HMI in vehicles, with particular focus on those that contain information and experiments with the use of HUDs. It also presents the state of the art in this field, namely what HUDs are available or in development for a PTW driver, as well as alternatives to HUDs that achieve the same goals.

2.1 Past studies and experiments

Most available literature on HUDs doesn't focus on PTWs, but on automobiles. Driving a four-wheeled vehicle is, in some aspects, very different from riding PTWs. However, in the particular context of HUDs, potential advantages will probably apply and even be enhanced, since, theoretically, drivers will always keep their focus on the road ahead, thus contributing for a safer driving experience.

Currently, there are eye gaze tracker systems, such as the JANUS system, that are able to accurately measure eye movements to calculate the time that the driver keeps his eyes off the roadway. These are accepted as a valid measure of drivers' distraction periods (Schweigert 2004). Distractions measured with eye gaze trackers occur when the person driving performs so-called secondary or terciary tasks (tasks that are not directly related to the act of driving itself, e.g. changing the radio station or checking the navigation system), which he/she performs by shifting his/her attention back and forth between the road and the HDD, resulting in periods of 1-2 seconds at a time without focus on the road ahead (Ablassmeier et al. 2007, 2251). It has been proven that HUDs can decrease driving workload and raise situational awareness by a factor of 0.2s, in comparison to HDDs (Horrey, Wickens, and Alexander 2003). Although this number might seem small, it has a big impact in drivers' attention level and may be decisive in critical situations.

There are a number of types of information that can be more easily and quickly assimilated through the use of HUDs. One of them is a warning sign that appears in the case that the safety distance between the driver's vehicle and the vehicle ahead is not being respected (Alves et al. 2013, 6). Although safety distance measurement is not in the scope of this dissertation, a possible future implementation of this feature might be interesting. Another warning signal might be used to control speed, by going off if the speed limits are exceeded. One study implementing such signals obtained positive results, with drivers surpassing the legal limit only 62% as much as drivers without the input signals (Doshi, Shinko Yuanhsien, and Trivedi 2009, 92).

HUDs may also imply some disadvantages in their use. For instance, a phenomenon called "perceptual tunnelling" may occur, causing the driver to lose sensibility in detecting objects in his/her peripheral vision. Another issue that might arise is distance overestimation, if information is projected in such a way that the projection is done using augmented-vision, on the windshield and onto the road (Ablassmeier et al. 2007, 2251). Of course, in the case of this project, information will not be displayed in such a manner, but on a monocle located to one side of the driver's facing position, so in principle, these disadvantages will not apply. Nevertheless, they are aspects to take into consideration.

Another difficulty that arises is that of achieving an optical contrast that allows a clear visualization of the displayed information. Thus, the light conditions play a central part in the efficacy with which information is perceived (Gale et al. 1998). Again, this has been observed for HUDs that are projected onto the windshield of an automobile. In the context of the application to be developed and the hardware to be used as the HUD, the screen will not be exposed to direct light, as it will be facing against it. However, light conditions will still affect the user's perception of the screen, to such a degree that they must be taken into account and tested in different conditions.

Moreover, background complexity also plays a role in HUD legibility, as well as the position in which the HUD is placed in relation to the driver's perspective. One study that projected the information onto the windshield of a driving simulator indicates that HUDs improve information perception with a low complexity scene in the background, while HDDs had better results for middle and high complexity scenes (Ward, Parkes, and Crone 1994, 460-461). These results may not apply to the problem at hand because, although information projected in the windshield blends and may be misinterpreted with middle/high complexity scenes in the background, Snow2's HUD is displayed in a screen that does not have a transparent background. In any case, it is to be expected that increasing background complexity will naturally hamper the legibility of any HUD, even if this issue does not prove to be so alarming in PTWs (with the proposed HUD system) as it appears to be in automobiles.

To sum up, practically all advantages found in the analysed literature do or probably will apply to the use of an HUD in PTWs, while most disadvantages either do not apply in the context of the project or will have a lower impact than in automobiles. Thus, utilizing a HUD for a PTW ADAS represents an opportunity with great potential that can greatly impact driving in general in a positive way.

2.2 State of the Art

2.2.1 Alternatives to HUDs

In today's market, to achieve the goals of navigation, communication and performance monitoring, there are significantly more alternatives to HUDs available than HUDs themselves.

One example are Head-down Displays (HDD), which have the same functionalities of an HUD but are located on the handle or some other part of the motorcycle, forcing the driver to divert his attention from the road for consultation or interaction. This process occurs repeatedly in order for the driver to fully capture the information being shown, forcing him to constantly lose sight of the road ahead for seconds at a time (Todoriki et al. 1994, 479). PTWs usually don't come equipped with HDDs, but these can be mounted using a GPS device, a smartphone or some other type of system.

The expansion of smartphone use has allowed for it to become one of the best and most easily obtainable auxiliary equipments for PTW drivers. Whether it is mounted on the vehicle or not, the number of mobile applications dedicated to improving driver experience is ever growing. Some examples of functions are the (normally) built in navigation system, traffic information or performance monitoring. These functions may either be accessed during driving via the mounted smartphone, headphones or speakers inside the helmet, or prior or after driving.

2.2.2 Current HUDs

Although this type of technology has only been available for a short period of time, some HUDs for PTWs are already in development and are even being commercialized.

The most advanced example is SKULLY's AR-1 helmet, which has an incorporated HUD in form of a screen located in the lower right corner of the helmet. Its features include navigation assistance, connection with the driver's phone (for voice-controlled calling) and a wide rearview camera that displays its video directly into the HUD (SKULLY 2015b). The first and only pre-order so far is estimated to arrive to customers in July 2015, but implies a cost of $1499 (SKULLY 2015a).

BIKEHUD has been the first HUD to be in the market and consists of a monocle placed on the helmet, a GPS sensor mounted on the handle bar and an onboard computer that must be wired into the PTW's ignition circuit. The monocle must be connected to the computer via a cable. This requires a non-trivial installation of this system. The features offered by BIKEHUD consist of basic information display such as user speed, time or virtual gear position, and optionally a very simply designed navigation system. Regarding their layout, this company abides by the “rule of three”, in which no more that three distinct pieces of information are shown, so as to avoid overloading the driver. BIKEHUD’s price starts at €409 (BIKEHUD 2014).

The project concept which most closely resembles this dissertation’s is that of NUVIZ’s HUD: it consists of an HUD which can be attached to any full-face or modular helmet, and is connected to an application on the user’s smartphone. Features include a navigation system, communication, and display of relevant information such as weather forecast, among other functionalities (NUVIZ 2015a). This project was crowd-funded and its only pre-order will only be available to those who contributed to its funding, having no future pre-orders planned (NUVIZ 2015b).

2.3 Conclusions

To sum up, not many literary documents can be found regarding this very specific problem and solution. HUDs have been employed in commercial automobiles since 1988 by GM (Changrak et al. 2014, 2277), and studies of their use in cars have been done for quite a few years, but they are mostly independent from one another and thus, no uniformization appears to be naturally happening, be it in terms of layout, displayed information or perceived benefits. Also, some internal and external factors influence the efficacy and usability of HUDs, like the position of the screen, the distance between the screen and the eye(s), luminosity and screen brightness, background complexity, and other aspects. Still, most research indicates there are advantages to using HUDs, specifically the ones mentioned in 1.1. In regard to motorcycle security, research supports the hypothesis that it is a much more dangerous way of transportation in comparison to cars, meaning that advantages brought by HUDs in PTW driving would have a bigger, more positive impact than on other types of vehicles.

Therefore, acquired knowledge on HUDs will be applied to a motorcycle HUD, with the expectation that similar positive results will be achieved, among others to improve driver security and enjoyment and the overall PTW driving experience.

Chapter 3

Software and Technologies

This chapter gives an overview of the software to be utilized for the development of the project, software requirements, the chosen IDE for the application development, and the external libraries, APIs and SDKs that will be used, as well as a detailed description of the HUD hardware that is used, Snow2.

3.1 Used Software and IDE

Two components will be developed that will be part of the system as whole, one to be run in the HUD and one to be run in the driver's smartphone as an auxiliary application. Both components will be developed in Java, to be run on the Android OS. Since the Android version the HUD supports is 4.1.2, the application to be run on that device has a minimum and a target Android SDK version of 16. The smartphone application's minimum Android SDK version is 17, and the target version is 22. Thus, Android Studio (AS) will be used as the IDE to develop both components. It has been chosen over Eclipse taking into consideration that AS is naturally more Android-focused and the official Android IDE, resulting in a slightly easier development process, and that its version (1.2.1.1) is stable enough to be considered a reliable IDE to work with.

3.2 External Libraries and APIs

Two external APIs are used in service of the HUD application and a third one is utilised for the smartphone application. All APIs are indispensable for the proposed value of the developed system, and their use requires a network connection.

3.2.1 Weather API

To acquire information regarding the weather, openweathermap.org's API for current weather data is used. The selection of the API took various aspects into account, specifically free usage, simplicity, and array of available information. The selected solution covers not only all current needs, but also offers other potentially interesting information for future development (OpenWeatherMap 2015).

3.2.1.1 HTTP request

All HTTP requests obey the following structure (blank spaces serve only the purpose of clarity and are to be ignored):

```
http://api.openweathermap.org/data/2.5/weather?
    lat=<LATITUDE>
    &lon=<LONGITUDE>
    &units=metric
    &APPID=<API_KEY>
```

3.2.2 Google Directions API

Regarding navigation, the natural choice was Google Directions API. Not only is it very thorough and complete, it also offers great documentation and support that is invaluable for the development phase (Google 2015b).

Both APIs mentioned above return responses in JSON format, that are filtered in the smartphone application and parsed and utilised in the HUD application.

3.2.2.1 HTTP request

All HTTP requests obey the following structure (blank spaces serve only the purpose of clarity and are to be ignored):

```
https://maps.googleapis.com/maps/api/directions/json?
            origin=loc:<LATITUDE>+<LONGITUDE>
            &destination=loc: :<LATITUDE>+<LONGITUDE>
            &waypoints=via:<LATITUDE>+<LONGITUDE>| (via...)
            &key=<API_KEY>
```

3.2.3 Google Maps API

Finally, Google Maps API is used in the smartphone application to allow visualization of performed tours, as well as the preview of new destinations to be added to the database. Its integration in the android application is seamless (Google 2015a).

3.2.4 Realm database

To implement the database on both the HUD and the smartphone side, Realm was utilized. It is a very simple mobile database that is not an ORM, but “is much faster than an ORM, and is often faster than raw SQLite as well” (Realm 2015).

3.3 Snow2

3.3.1 Introduction

The HUD that will be utilized for this project is Recon Instrument’s Snow2, shown in

Figure 3. 1 - ReconInstrument's Snow2 HUD

Figure 3.1. It is currently used for winter sports practice and was originally designed to be used inside snow goggles.

This device not only contains the screen itself to display information, but also includes GPS, various sensors, namely gyroscope, accelerometer, magnetometer, altimeter and thermometer, and Bluetooth and WiFi communication. The refered screen has a resolution of 428x240 pixels, and thus the HUD does not follow the norm, in the sense that most HUDs project information onto the windshield or whatever surface is in front of its user. The screen is designed to be placed in the lower right side of the user's field of vision, and its format does not allow for other types of device layout, be it in goggles or a motorcycle helmet.

To ensure HUD ease of use and simplicity, the aforementioned components play an important role, allowing the hardware to be used without having to physically connect it to the PTW, e.g. like BikeHUD's system. Because of this, information will not always correspond 100% between the HUD and the motorcycle (e.g. exact instantaneous velocity), but the differences will, in principle, be minor and will not have any relevant impact.

Regarding the interaction between the user and the system, the only way to directly manipulate the application running on the HUD is through a remote control that is also a component of Snow2. This remote is a bracelet with 4 arrow buttons, one back button and one OK button. It is waterproof and was built so that it is as easy to use as possible, even with gloves on. The communication between the hardware and the remote control is done with Bluetooth, specifically Bluetooth Low Energy (BLE), which, as the name points out, consumes a minimum amount of battery.

Snow2 allows connectivity with the driver's smartphone and also third-party devices such as cameras. This feature is critical to the development of this system. Connectivity is possible through Bluetooth or WiFi (direct).

Issues that arose during the development phase included how to mount Snow2 in a helmet and how to incorporate an external microphone and speakers to allow call management, since a built-in headset is not included in the hardware. All issues related to this device can be found in section 3.3.4.

3.3.2 Technical specifications

The following table indicates the technical specifications regarding Snow2's HUD.

ONBOARD SENSOR FRAMEWORK	9-Axis sensors • 3D accelerometer • 3D gyroscope • 3D magnetometer
	Pressure sensor • Altimeter & barometer application
PROCESSOR	1 GHz Dual-Core ARM Cortex-A9 with PowerVR SGX540

SUBSYSTEM	GPU On-board memory • 1GB DDR2 SDRAM • 2GB flash
ULTRA-COMPACT DISPLAY & VIRTUAL IMAGE	Wide screen 16:9 WQVGA ultra-compact display at 428×240
	Virtual image appears as 14″ HD display at 5′
	Power-saving sleep mode
	High contrast and brightness for readability in high ambient lighting
	High Color 16-bit (5:6:5)
NETWORKING SUBSYSTEM	Wi-Fi (IEEE802.11a/b/g/n) Bluetooth 4.0 (Bluetooth Smart) • Apple MFi Bluetooth support
	GPS • INS Kalman filtering fusion algorithm
	Micro USB • Device charging/power • Data transfer
WEIGHT	~65Grams (~2.3oz)
BATTERY SIZE AND LIFE	1200 mAh = 6 Hours+ Per Charge

Table 1 - Snow2 tech specs

3.3.3 Recon SDK

For native ReconInstruments applications (that run both on Snow2 and the company's other product Jet), ReconInstruments provides an SDK called Recon SDK that allows access to the built-in sensors and web connectivity. This promised to be more useful than it was in reality, taking into account that only instantaneous speed and bearing (with the compass) are obtained through the SDK. It is also very poorly documented and its usage is not very clear, which served as further discouragement to utilise it. At some point during the development phase, inclusively all documentation was removed from ReconInstruments' website, so that taking further advantage of Recon SDK proved impossible. This happened due to a new release of ReconOS and SDK, which will have new documentation, but has not been launched as of today. An attempt was made to retrieve that information through a contact with the company, but such was not possible.

3.3.4 Issues

Since the utilized device isn't designed for PTW use, but instead snow sports, some issues naturally arose concerning its adaptation to the project at hand.

Throughout most of the development phase, as well as the intermediate experiment, the device was used and tested in its original mount, namely a pair of UVEX snow goggles.

Figure 3. 2 - The Snow goggles w/ Snow2. 1 - screen. 2 - battery (not visible). 3 - charge plug

The first implication with the aforementioned setup is the fact that the screen is not actually at the user's eye level, but rather in his/her eyes' lower right corner, as can be seen in Figure 3.2. The distance from the driver's forward view to Snow2's screen is naturally much shorter than that to the handle bar or speedometer, but the level and possibly even the amount of the advantages offered by this solution is lower than those of a display at eye level.

Furthermore, the used goggle's lens color was orange, which changes the user's perception of what is in front of him. For ski or snowboard, orange is a useful color to serve as protection against sunlight and at the same time enhance depth perception in an environment that is generally very bright. On the street, however, it is one of the least recommended types of eyewear to use, especially for night drives. In the development phase(s), this was not a factor that deserved a high degree of concern, since the focus (UI and UX-wise) was on the information displayed on the screen, the display and placement of said information, among other issues (please refer to Chapter 5). The same may not be said about the experiment phase. Taking into consideration the real test environments in which it was performed, it is not possible to determine the extent of the influence that this variable has had in the drivers' recorded experiences. For more information regarding the performed experiment, please refer to Chapter 7.

Finally, an issue regarding the device's physical structure was identified. Snow2 HUD is designed to fit into snow goggles, as explained before. These have a very specific shape and limited space, so the hardware is placed in a very strategic way. The screen itself is at the lower right corner of the goggles, and the battery is placed on the left side. Both components are connected through a cable that goes along the upper edge of the goggles.

Figure 3. 3 - Snow2's structure w/ screen and battery connected by cable

Figure 3.3 shows the display of the whole structure. Because a motorcycle helmet has a very different shape, some thought had to be put towards the placement of Snow2 HUD in the developed helmet prototype, as well as possible alterations to it. All details about the implemented solution can be found in the next section, 3.3.5.

3.3.5 Helmet mount

A shop specialized in PTWs and motorcycle apparel was consulted regarding this matter, and three solutions were proposed to create a working propotype. The first option was not taken into serious consideration, but is nevertheless worth mentioning. The existing HUD mount, namely the goggles, would be used and incorporated into a specific model from the helmet brand SHARK, which has a special fit for sports goggles. This would not only be very restrictive in respect to the array of helmets that could be used to incorporate the developed solution, but would deem the goggles a necessity, which isn't a good option since many PTW drivers don't use them.

With the aforementioned proposal cast aside, either a built helmet would have to be adapted somehow in order to fit Snow2 HUD in a practical and logical way, or the helmet parts would be bought separately, altered should the necessity arise, and put together, which would allow a bigger customization of the helmet. Although both alternatives represented fair advantages, the decision was made to utilise a finished helmet and apply the necessary changes to it. Firstly, this option would represent less work, since the adaptation of a helmet is much simpler than building one from scratch, especially with the project's needs in mind. Secondly, the projects goals and objectives were taken into account: the solution should be as modular and adaptable as possible, and the creation of a solution-specific helmet would be a direct violation of that premise.

For the first prototype, helmets with a considerable amount of inner padding were taken into consideration. This is because the hardware has to be attached to the inner part of the helmet's outer shell, and helmets with pads have more space for it.

One goal during the prototype creation was to place the HUD more at the users' eye level, in order to maximize the advantages discussed in Chapter 2. Of course, this could not be intrusive to the driving activities; the screen would just be placed higher vertically, but would not move horizontally, so that drivers would only have to slightly direct their attention to the right and not to the lower right corner of their fields of vision.

The selected helmet was an NZI Verti, a modular helmet with the aforementioned amount of inner padding. The choice was made to acquire a modular helmet not only to be able to simulate an integral helmet and an open helmet with just one prototype, even if in a very general manner, by opening and closing the front visor, but also because it was practical in terms of building it and fitting all the single components. Velcro strips with glue on the other side were used to fix ReconInstrument's hardware device into place, with the advantage that it may be removed and placed again very easily and also adjust its position to the driver's preference. The HUD's cable was fixed into position with tape. After experimenting with the best placement for the HUD's battery, taking into account its relatively big size, it was decided it should stay on the outer part of the helmet shell, on the right side. In any other way, either some part of the helmet (e.g. the padding) would have to be cut off, or it would be a big discomfort for the driver to have so much pressure applied to his/her head, caused by the battery in the interior part of the helmet. The Bluetooth device that connects to the headset is also placed on the outer part of the helmet, on the left side, and stays in place with a latch that can fit into practically any helmet model. Both these components being placed on the outer part of the shell implies having cables run from the inner to the outer part of the helmet. For the Bluetooth piece, this caused no issue since the cable was long enough and there was a margin for placing it in an appropriate position. As for Snow2's cable, its length is very reduced, resulting in the right side's padding not fitting like it should.

Figure 3. 4 - Prototype w/ visible microphone, BT device and HUD

Figure 3.4 shows the Bluetooth piece, the microphone and the HUD all built into the NZI helmet. The missing padding on the right side is also visible. In order to place it again in a way that is practical and at the same time comfortable for the driver, the padding would probably have to be modified and some parts would have to be removed in order for it to fit.

Regarding the HUD's position, a focal point of the prototype design, it was attached onto the side of the helmet, right at the level of the cheekbone. This was considered the best position for the device, taking into consideration the possibilities available for successfully postioning it inside the helmet. As a result, the initial wish to elevate it to eye level was unfortunately not possible to achieve. Moreover, the screen is the slightest bit tilted, which is noticeable but negligible, as the adaptation is quite simple. Finally, from where the HUD is positioned, the screen's arm is not long enough for the screen to be visible with both eyes, similar to when it is used mounted on the original snow goggles. Still, because the screen is not surrounded by a dark colour like it is by the padding of the goggles, it is slightly less clearly seen on the motorcycle helmet because of the smaller contrast that exists.

3.3.5.1 Conclusions

During and after the prototype development process, several conclusions were reached. First and foremost, the modularity this project aimed to achieve is not an easy task, and adapting the current solution to any other helmet, especially other types of helmets (integral, open), or using a different BT headset, would not be a straightforward task. Since a considerable amount of space inside the helmet is required, and several different components come into play, every case would be at least slightly different, and a general analysis of how the solution as a whole would work would have to be analysed each time. With a high degree of probability, the result would be similar in most cases, but the desired ease of installation, at least, is not as intended at the start of the project.

Also, because the BT headset is independent from the HUD, both equipments require an initial setup to be paired to the smartphone via BT, and must be charged separately. Charging the BT headset presents no problem at all, since the BT device detaches from the helmet easily, but could be an issue for the HUD. Not only is the device's cable taped to the helmet (this is not essential, but it's the case for the prototype), but the device itself would also be practically completely covered by the padding. This would impede an easy cable connection in the case that the HUD could not be removed. To solve this, a short cable could be always connected to Snow2 and easily tucked into the paddings of the helmet, and could be connected to a charger whenever needed. Nevertheless, having to carry the whole motorcycle helmet to the charging place is not practical and preferably avoided by not using tape or any other permanent fixing materials (tape is not permanent, but its illogical to remove the old tape and use new one for each charge).

As ReconInstrument's Snow2 is built specifically to fit into snow goggles, its design is fairly restrictive. Ideally, a version of Snow2 would be built to be more motorcycle helmet friendly (and other types of helmets). Two major factors would have to be addressed. Firstly, the cable connecting the screen and the battery should be longer, to allow more freedom when placing the battery inside or outside the helmet, because the screen would always have to be placed in the same position. This could ultimately even have positive impliations in the placement of the screen itself. Secondly, the arm that connects the screen to the device should also be longer, with more rotation being possible for a better adjustment to the drivers' preferences. In a best-case scenario, a whole new HUD design would be implemented by ReconInstruments to perfectly adjust to any kind of helmet, or ate least specific types of helmets, with a BT headset integrated into the system for maximum practically (in the sense of less independent components, easier charging mechanism, among other aspects). However, such a solution is both not expected and not in the scope of this project.

Chapter 4

System Architechture & Modules

4.1 System Architecture

4.1.1 Overview

The whole system is made up of several components that communicate through various ways. The components consist of the HUD application running on the Snow2 device (that originally mounts on ski goggles but later in the project development was mounted on a motorcycle helmet), the auxiliary smartphone application running on a smartphone, a remote control to give inputs the the HUD application, and a headset complete with speakers and microphone.

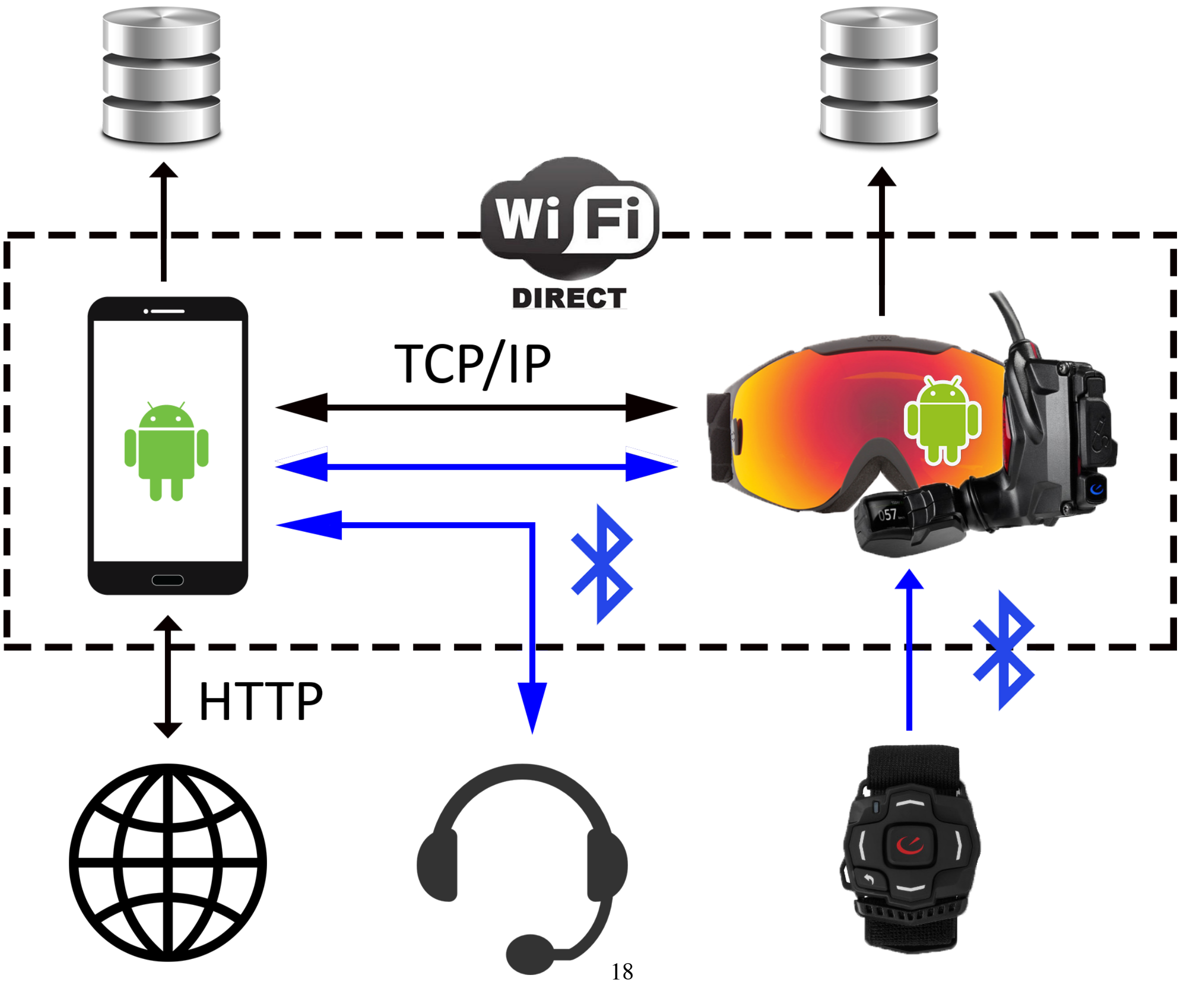

Figure 4. 1 - System architecture diagram

Figure 4.1 shows the general system architecture, complete with communication (form and direction).

4.1.2 Communication

Communication between the HUD and the smartphone is done in three different ways and serve various purposes. It is invaluable to the global value proposed by the project as a whole, since ReconInstruments' device is not capable of performing all required tasks, most importantly establishing a network connection that allows the various API calls. It is also not suited for other kinds of functionalities, which include the visualization of marked tours, for instance, for obvious screen size restrictions (and also the lack of network connectivity). Other aspects may prove themselves relevant for a continued application use, for instance the storage space that a smartphone offers and that the HUD doesn't.

4.1.2.1 Bluetooth communication

The two devices can be connected via Bluetooth using Snow2's native connectivity system, which offers more functionalities than the ones used in the scope of the project. The previous version of ReconOS, version 3.2.1, included a music player that the current version (4.0) does not (only through third party applications can the same result be achieved), and allowed the user to select which music to play, as well as basic player functions like play and pause.

What is truly essential in this form of Smartphone-HUD connection is the possibility to perform all incoming call handling, as well as receive text messages via notifications in the HUD. For more information on this matter, please refer to Section 4.2.5.

4.1.2.2 WiFi direct communication

WiFi direct is used only as an indirect means of communication, and its only purpose is to create a local network that the devices may join and allow both to establish a TCP/IP connection, providing them with the other device's IP.

This type of communication offers various advantages, for instance compatibility between different devices, types of devices and devices from different manufacturers, an essential characteristc for this project. Additionally, connection speed is very fast, faster than with Bluetooth technology, and distance across the devices may also be longer than with BT. The fact that mobile data and other wireless connections are not interrupted is also important, because some HUD requests take advantage of the smartphone network connectivity.

When the HUD activates WiFi direct (this occures automatically in activities that require connection with the phone) and connection with the HUD is requested by the phone application, the user must accept the incoming connection invitation in the HUD to successfully establish a WiFi direct connection. After both devices are within the same group/ network, the group owner is the HUD, whose IP address the phone now has access to. After this process, the TCP/IP connection is established. The WiFi direct connection, as well as the TCP/IP connection, must only be established once, while both applications are running.

4.1.2.3 TCP/IP communication

The HUD application serves as the TCP server and the phone application as the TCP client, so that only the phone requires the server's local IP address to request a connection with the server. In turn, the server listens for incoming connection requests until the first one is received, since only two devices will connect and thus there is no need to maintain a thread for the purpose of listening for other requests that would never arise. The used port is 7077, by default.

After the TCP/IP connection is successfully established, both the server and the client keep a thread running for message receiving, and send messages as necessary, depending on the user's activity with any of the applications.

4.1.2.4 TCP/IP communication protocol

A messaging protocol was defined for all messages exchanged between the applications. All message templates described in this section are the ones received by the associated device. They correspond to messages sent by the other device. Most of the messages are in String format, but a few are exceptionally sent in JSON format. The first token of any message is always a description of the message, and serves practically as an ID that the other device recognizes, reacting accordingly. All other tokens, if any, are parameters that contain all information the receptor requires.

HUD

- SYNCHRONIZED_DESTINATIONS

- DIRECTIONS <DIRECTION_STEP>;<DIRECTION_STEP> …

    - <DIRECTION_STEP> - <ORIGIN>!<DESTINATION>!<STEP_DURATION>!
      <STEP_DISTANCE>!<STEP_DIRECTION>

- WEATHER <WEATHER_INFO>

    - <WEATHER_INFO> - JSON Object that contains “temp” [temperature in ºC], “humidity” [in %], “windSpeed” [in km/h], and a JSONArray <WEATHER_ATTR>
    - <WEATHER_ATTR> - JSON Array that contains “id” [id of the weather type according to API service], “description” and “icon” [string that indicates which icon to display]

- CONTACTS <FULL_NAME>||<PHONE_NR_ARRAY> <FULL_NAME>|| <PHONE_NR_ARRAY> …

    - <FULL_NAME> - <NAME>__<NAME>(__<NAME>...)
    - <PHONE_NR_ARRAY> - <PHONE_NR>__<PHONE_NR>(__<PHONE_NR>...)

- METRICS <MAX_SPEED> <TOTAL_DISTANCE> <TOTAL_DRIVE_TIME>

Smartphone

- REQUEST_DIRECTIONS <ORIGIN> <DESTINATION> <WAYPOINT> <WAYPOINT_COORDINATE>

    - <WAYPOINT> - {WAYPOINT, NO_WAYPOINT}
    - <WAYPOINT_COORDINATE> only exists if <WAYPOINT> has WAYPOINT value

- DIRECTION <DIRECTION>

    - <DIRECTION> - {LEFT, RIGHT, UTURN, STRAIGHT}

- SYNCHRONIZE_DESTINATIONS
- REQUEST_WEATHER <LOCATION>
- BEGIN_TOUR <TOUR_TITLE> <TOUR_TIME> <TOUR_DISNTACE>
- TOUR_POINT <POINT_NR> <POINT_COORDINATES>
- END_TOUR
- REQUEST_CONTACTS
- PHONE_CALL <PHONE_NR>

    - <PHONE_NR> may or may not have indicative

4.1.3 GPS Location

In order to obtain the user's GPS location, the HUD's included GPS sensor is utilised. Android's LocationListener is used, with location updates requested with a minimum update time of 500ms. Unfortunately, the Google Location Services API could not be used, even though Google stronly advises it due to better accuracy, a simpler API, among other aspects. The reason for this is Snow2 does not support Google Play, so the API, which is a Google Play Service, cannot work. The LocationListener implementation is instanciated only once, when the HUD application is started, and the device's constantly updated location is available to any other class or activity, provided there is a GPS connection. Such connection is easy to obtain, no special settings are required and it is automatically established if the device is in an open area.

4.2 System Modules

4.2.1 Weather

Figure 4. 2 - Screen captures from the weather module (HUD application)

To obtain information on weather and temperature (Recon SDK supposedly offers temperature readings, but it was concluded via research that it does not work properly), a free Web API is used, provided by OpenWeatherMap. Although a variety of options is offered, such as historical data from cities, data from weather stations and weather forecasts, only simple requests for current weather data are made. These requests are only made regarding the user's current location, but could easily be utilised for a destination in case of a navigation request, or for intermediate points in a specific route.

The JSON response to the weather API request that is currently being made contains a very complete and diverse array of information. Following the principle that only relevant data should be presented, the final filtered parameters are temperature in ºC, humidity in percentage, wind speed in km/h, a weather description and the related icon's name, to then select the correct icon to display in the HUD (all weather icons are stored locally in the application).

To obtain weather information, HUD GPS connectivity must exist, as well as a TCP/IP connection with the smartphone application (meaning it must also be running) and network connectivity with the smartphone. In the main menu, the Weather activity should be accessed and the API call is automatically conveyed to the smartphone, which in turn makes the request and gives the response back to the HUD, provided the mentioned conditions are all true. In the same Weather activity, the current weather description and a corresponding weather icon are shown. If the user then scrolls horizontally or accesses driving mode (and is not being navigated to any destination), data regarding humidity, wind speed and temperature are displayed.

4.2.2 Tour tracking

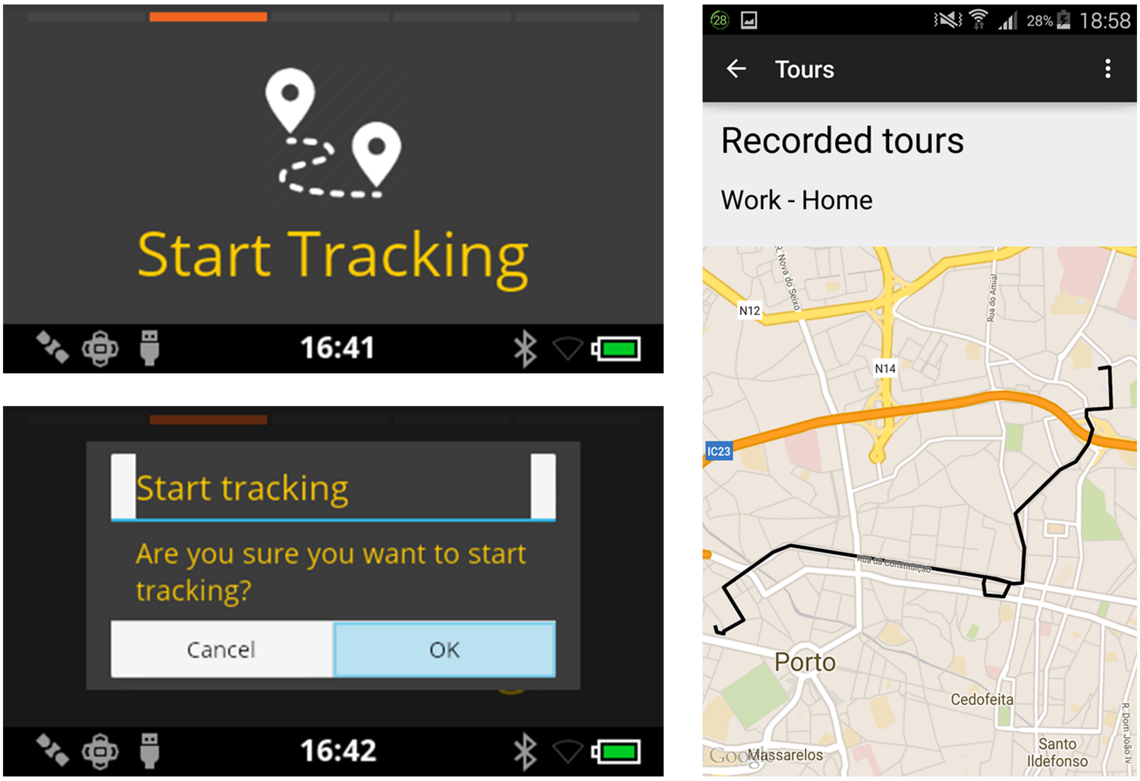

Figure 4. 3 - Screen captures from the tour tracking module (HUD and Smartphone applications)

One of the main features of the system is the possibility to record tours and posteriorly view them and even repeat them. To achieve this, GPS location updates are analysed and stored in an array as long as the user is tracking his or her route. After finishing the tour, all location points associated with the tour are stored as tour points in the database, along with total tour duration and length.

Stored tour data is not relevant for the HUD application, and is stored only until the user requests the tours' synchronization with the smartphone application. As can be deduced from the communication protocol in Section 4.1.2.4, points are sent one by one from the HUD to the smartphone. Even though TCP/IP communication is very reliable and rarely any error occurs, tour points are numbered as an extra safety assurance. Point order is important when tours are being displayed, considering Google's Maps API is used for the display. Specifically, this API's Polylines take point order into consideration and connect all points sequentially.

After the synchronization, all tours are stored in the smartphone application, where a user may change tour names or delete tours. Also, he may visualize all tours on the map. Additional tour information about total tour time and tour distance is available, which is valuable simply as sheer tour related data, but also in the case where a user whishes to compare tours, for instance to select the optimal route to a specific destination, or to analyse tour times over time or with varying traffic levels.

As far as the amount of tour points stored is concerned, location updates are very frequent, and a fine precision results in points in a very close proximity being added, especially if the driver is travelling at slow speeds. Taking this into consideration, storing points with only 1-2m distance from each other for a tour that can be hundreds of kilometers long may result in a very considerable amount of database entries. Figure 4.2 demonstrates a very short, 5km long tour that was marked using the implemented tour tracking mechanism. The view is zoomed out, but still the impressive amount of points that was stored can be perceived. This becomes even clearer when zooming in (Note: the display of this tour was not achieved with the smartphone application's implemented tour visualisation, as it demonstrates tours as polylines and not as sets of points).

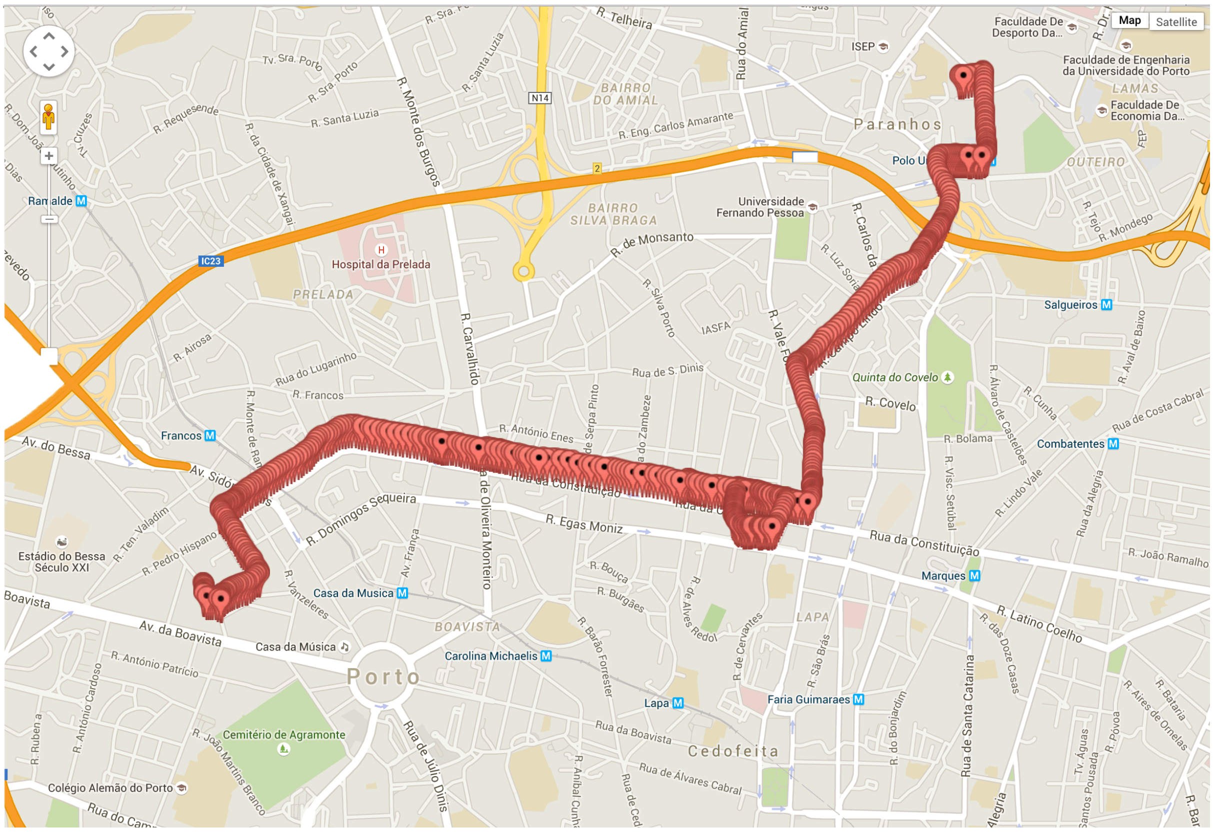

Figure 4. 4 - Example of a tour, with all single points marked on the map

Even this being a key component of the system as a whole, very in depth development of any one system module was not possible due to time limitations. Namely, there are various algorithms that serve not only to improve point precision, by use of additional sensors for instance, but also to reduce the required amount of points stored in order to be able to reproduce and visualize the tour without loss of precision. A very simple filter was implemented that by itself represented a big advantage. The filter consisted of only storing points that were at least 3m away from the last stored point. This takes most effect in the moments where the driver is moving slowly and thus very little, for instance when waiting for a traffic light to turn green. Clusters of unnecessary points are this way reduced to two or three points, that are enough to maintain a very good point marking precision. Applying this filter to the example tour of figure 4.4 resulted in a reduction of 665 points down to 604 points. This represents roughly a 10% decrease of stored points. As 600 points for a 5km tour might suggest, being able to reduce 10% of all stored entries with such a simple filter brings a huge benefit especially for big tours. In the future, other, more complex filters and algorithms may be applied to obtain even more positive results.

4.2.3 Driving mode

Figure 4. 5 - Screen captures from the driving mode module, w/ all 3 variations (HUD application)

As previously mentioned, this system does not connect directly to the PTW, so that no On-Board Diagnostics (OBD) data is collected. This simple fact implies some limitations to what driving metrics and other useful information may be collected and analysed (for instance the precise amount of gas in the tank or the current gear), but Snow2's incorporated sensors and GPS fill this gap, at least partially.

The basic information that is always displayed, provided there is gps connectivity, is instantaneous speed that is updated every 0.5s and is obtained through Recon SDK. Other simple but important information is saved whenever the HUD application is running, namely total covered distance and total traveled time. In a future development phase, time should only be taken into consideration when movement is detected, and stop being taken into consideration after the driver is stopped for a certain period of time. Total elapsed time is an information relevant in itself, but especially relevant for calculating average speed. Average speed at any time is total covered distance / total drive time. For tours, average speed is also available by dividing one tour's total distance by its duration (for more information, please refer to Section 4.2.2). Currently, these are the only recorded/ available metrics.

In the future, other more detailed parameters are planned to be calculated. A method of calculating the gas level at any point would require some user input and would consist of measuring the average distance covered between refuellings, as well as registering the amount of fuel bought on each refuelling. Combining this data, some formula can probably be created to obtain average fuel consumption, gas level and possibly other aspects, e.g. average fuel consumption over time as a basis for advising the driver on whether he should be a less aggressive driver (if that average rises over time).

Of all the aforementioned metrics, only instantaneous speed is displayed on the HUD, all others being available on the smartphone application. But the Driving Mode activity displays other types of information, not necessarily related to driving performance. Depending on the information available on the HUD, three variations of the activity may exist (see Figure 4.5). The basic one can only display instantaneous speed in km/h, as no API calls were requested. In the case that a request for weather information has been made, the activity will also contain information regarding wind speed in km/h, humidity level in %, and temperature in ºC. The driving mode variation that has the highest priority will be displayed if a navigation request was made and the user is in the process of receiving navigation input. Over the instantaneous speed, a medium-sized direction arrow corresponding to the next action/turn to be taken is visible, with the distance from the current position to the turn's position presented below in km. In this manner, the driver can be much more aware of the path that must be followed, and will be given more time to react and follow the instructions, given in a visual and audio manner. For more information regarding navigation, please refer to the next section (4.2.4).

4.2.4 Navigation

Figure 4. 6 - Screen captures of the navigation module (HUD application)

Navigation is a key feature and a potentially very useful tool for PTW drivers. All navigation input cannot be as complex as in a smartphone application or a car's onboard computer due to obvious screen size restrictions, so the key to implementing this feature on an HUD of Snow2's nature is to find the essential information that must be conveyed, and do so in the most clear and intuitive way possible.

The decision was made to take advantage of Google's Directions API, for several reasons. Firstly, Google's seal of quality is a great reassurance and translates into a quality service that probably no other API could offer. Its use is also free of charge, also a priority bearing the projects' goals in mind. Furthermore, some response contents are very useful, not considering the main and basic direction information. Finally, it is very well documented, allowing for a very supported development phase.

For the navigation function to successfully work, both a network and a GPS connection are required. After accessing the navigation activity, all destinations stored in the HUD are listed, and a favourite destination, a recent destination or a tour may be selected. After the destination selection, a message is sent to the smartphone via TCP/IP containing the driver's current GPS location, as well as the destination's GPS location. In the case of a tour, a waypoint (also a GPS location) must also be included in the message, and corresponds to the tour start, and the destination location corresponds to the tour end. The received information is then used by the smartphone to make an API call. To cancel navigation indications, the driver must long press the back button while in the navigation activiy.

The response for any directions request is in JSON format and contains a great deal of information, not all of which is relevant to what can and should be shown on the HUD. Thus, the response is parsed and all essential data is sent to the main application.

A thread is then started that iteratively processes all direction steps and gives the associated navigation inputs accordingly. All these steps contain a starting location and a final location, the latter being used to determine wether or not a certain direction step has been taken and another may be started. A distance threshold of 100m or lower was determined as acceptable to advance to the next step. Since the navigation input is only given after any one step is concluded, this threshold takes into consideration the fact that a driver needs to be warned beforehand of the next turn to take, with an amount of time that allows for safe and secure reactions. In fact, with a driving speed of 100km/h, the threshold only allows the driver ~3.6s to react to the navigation input, resulting in a missed turn or a dangerous maneuver. Nevertheless, it is expected that the driving mode activity will be active during practically all driving period, containing the next turn's direction and the distance remaining to said turn. With that information, the driver's reaction time will be much longer. Also, there are two additional factors to take into consideration when determining the referred threshold, which may not be too high. Firstly, the distance between the current location and the next intermediate point is calculated as a straight line and not as a line that goes along the road. In this way, a very large threshold could, in some cases, allow the input to be mistakenly given before it actually should be (e.g. if a road is naturally sharply curved). Secondly, in cases where there are several roads in short succession that lead in the same direction as the one the driver should take (e.g. cities), a large threshold could lead the driver into taking the turn in the correct direction, but the wrong road.

A navigation input is given via an activity that is run no matter the state of the application, and very simply contains a directional arrow that can point in one of four directions: up meaning forwards, down meaning backwards (u-turn), left meaning a left turn and right meaning a right turn. One exception to the navigation input activity content is in the case of the last direction step, where the driver has reached the destination. In this case, an icon indicating the arrival to the destiation is displayed. The other exception occurs when a mistake is made, triggering the appearance of a cross to indicate said mistake. For more information on the general activity layout and appearance, please refer to chapter 5.

Apart from a visual input, an audio command is also given through the speakers. Both inputs complement each other and work as a unit. In total, there are currently 5 voice prompts being used, namely:

- "Turn left"
- "Turn right"
- "Straight ahead"
- "Make a u-turn whenever possible"
- "Wrong way. Calculating new directions"

Whenever a navigational input is given, a message is sent from the HUD to the smartphone containing the respective sound command to be given. The smartphone then plays the sound command through the headset's speakers, which should be paired via BT.

The user isn't aware of all the direction steps that are parsed. Namely, there are some steps that do not indicate a turn, but rather an indication or information. Mostly, these indications refer to the lane or position the driver should be in, as a preparation for future turns. The same directional arrows cannot be used for both turns and the advised position on the road, so these steps are currently simply ignored in the sense that no input is given. In the future, some way to also give feedback on these steps will also be thought of and implemented. The API's response already has an indication of which steps are turns (and in this case the direction of the turn) and which are not, a useful information also in terms of the navigational arrow orientation, since those indications specify the type of action (turn, u-turn, ...).

Because the Directions API does not take the user's bearing into consideration, some way to give the driver the first indication had to be determined. This can be achieved in two ways. The first is possible only if there is a GPS connection and at least one location has been collected. Android's Location class contains a function, *bearingTo(Location location)*, that is used to calculate the bearing between the user's current location and the final location of the first available driving step. This bearing is a degree value (East of true North) between 1° and 360° (0° is represented by 360°). The first indication will be forwards if that degree value is [315°-360°] or [0°-44°], right turn if it is [45°-134°], u-turn if it is [135°-224°] and left turn if it is [225°-314°]. In case there is no GPS connection, which can happen if the driver is for instance in a garage, Snow2's compass value may be used to achieve the same goal. Using this attribute, it is possible to calculate the first navigation input by comparing it to the cardinal direction that should be taken initially. This direction is calculated using the first direction step's starting and final location. The difference between the bearing given by the device's compass and the cardinal direction will result in a degree value that is used to calculate the navigational arrow's direction in the matter explained above.

As mentioned before, all directions are fetched once and stored locally in the HUD, so that navigation is not dynamic. This constitutes an issue that must be overcome in the case that a wrong turn or some other error is made. Although there might be exceptions to this solution that are not correctly handled, it is assumed that an error was made if the distance between the driver's location and the next intermediate location grows for a certain period of time. Should such an error be detected, the thread handling the direction indications is terminated, as new directions must be requested. Naturally, the smartphone must be running the application, the TCP/IP connection must be established and a network connectivity must exist (if the driver is on the road, this would mean mobile data would have to be enabled). If at least one of these requirements is not met, an automatic route recalculation cannot be completed and the user would have to manually request new directions. Even if new directions are successfully received by the HUD, the possibility exists that these will already be obsolete, if the driver is moving during the direction request period and already went by the place where he should have turned. This is not critical, and actually happens in other GPS systems as well.

4.2.5 Call handling

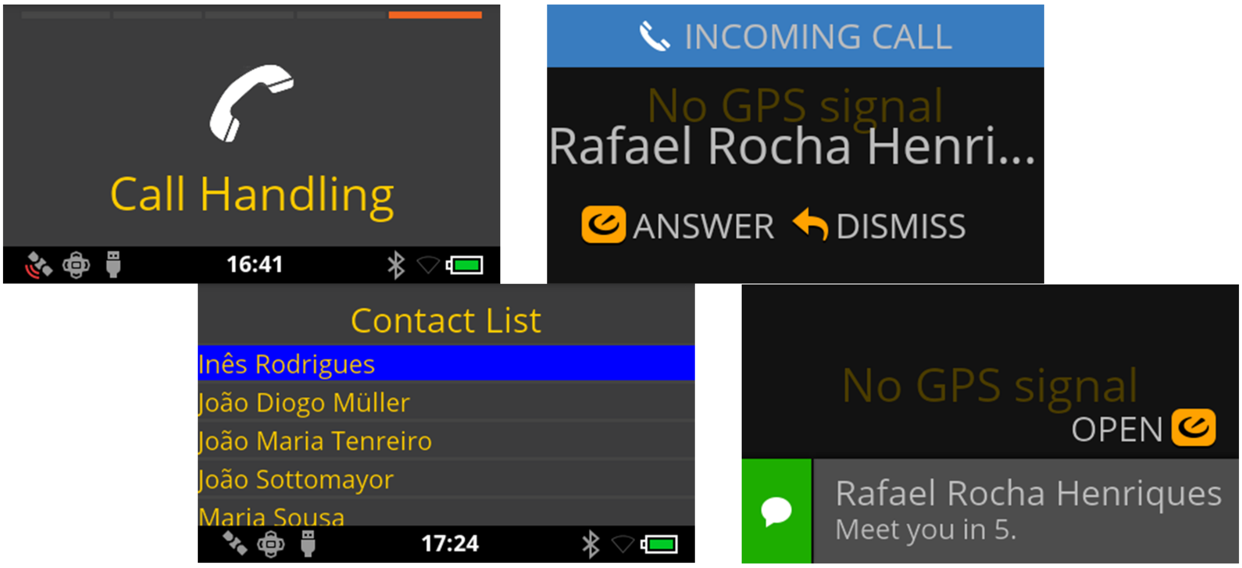

Figure 4. 7 - Screen captures of the call handling module (HUD application)

Call handling is fully supported by the HUD, both natively and with the developed application, and partially supported by the specific BT headset that is used for the prototype.

As long as the smartphone is paired with BT to Snow2 and BT notifications are enabled, incoming calls and text messages are received and displayed on the HUD. Call acceptance or rejection can be controlled with the remote control, although this feature is currently not supported for smartphones running an Android version of 5.0 and above.

Because Snow2 doesn't have speakers or a microphone, the BT headset is used to transmit audio output and collect audio input. Incoming calls are accepted by it automatically after 3 to 5 seconds, if no other action is taken. Rejecting calls is also possible by clicking on the BT device's power button.

Thus, incoming call handling is supported in 2 various ways, but making outgoing calls is only achievable through the application. Every time the HUD application is started, all contacts are requested from the smartphone the first time the call handling activity is opened. Firstly, though, the TCP/IP connection between both devices must be established. After the contacts are received, they are listed and shown to the user that can scroll and choose a contact to call. If a call request is made, a message is sent to the smartphone with the selected contact's phone number, and a call is made with it.

To end an active call, the remote control's back button must be long pressed. Alternatively, the BT device's power button can also be pressed. Speaker volume may only be controlled with the headset's volume up or volume down buttons.

Chapter 5

Layout design (HUD)

Layout design is a core aspect of the project at hand. Firstly, the HUD's screen is of very small dimensions, specifically with a resolution of 428x240 and a permanent landscape orientation, which leads to the need of a very organized user interface, in order to convey the perfect amount of information and avoid overflow of data or waste of screen space. Secondly, in addition to the amount of information, the form in which it is conveyed is critical to ensure driver security and system efficacy, taking elements such as screen brightness, component sizes, colors and component disposition, among others, into account. Layout components must be well thought out and designed to achieve the aforementioned goals. This chapter describes all aspects regarding layout specifications, needs and design.

5.1 ReconInstrument's design guidelines

Recon Instruments provides a set of design principles and patterns which, in theory and for the ReconOS platform, maximize user experience. These have been taken into consideration and compared with information gathered in the literature review, with a prospect of achieving the best possible results regarding information transmition and security. They have been largely adopted, as the company no doubtebly possesses years of experience in this field, with its first HUD released to the market in 2010, and has a vast amount of expertise designing for a type of screen that is not very widespread and is not extensively studied.

5.2 Margins and spaces

Margins of 20px are used along all four sides of the screen. Although a specific reason for this recommendation is not given, it is very probably because the edges of the screen are not as clear and focused as the center. Since a magnifying glass is used to enhance the pecepted screen size, the screen edges often appear blurred, and should therefore not contain any information at the risk of the user not being able to correctly identify or interpret them. Also, it is good practice to have an uncluttered interface, with generous margins throughout, in order to facilitate navigation and decyphering at a simple glance.

5.3 Typography

In regard to typography, the ReconInstrument's default font, which was kept, is Open Sans. For text in the main menu, a font size of 40sp was chosen; titles in modules' main activities have a font size of 30sp, and list items (e.g. destinations, contacts) have a font size of 22sp. Lastly, medium-sized text, such as the one displayed in driving mode, has a font size of 25sp. The difference between titles and normal text is not very significant, because even the small items must be easily identifiable and readable. This relation of sizes was chosen to achieve not only that, but also to distinguish the importance of shown text in a simple manner – a title will be identified as such with no difficulty whatsoever.

5.4 Colors and layout versions

In total, three different versions of the layout are offered, and each has its own set of colors. Each set is detailed below (all color codes are in hexadecimal format):

Layout 1

Background color:	#3C3C3D	EXAMPLE
Text color:	#FFCC00	EXAMPLE
Icon color:	#FFFFFF	EXAMPLE

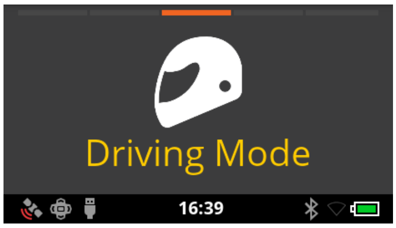

Figure 5.1 – Layout V1 example

Layout 2

Background color:	#EDEDED	EXAMPLE
Text color:	#3564FC	EXAMPLE
Icon color:	#000000	EXAMPLE

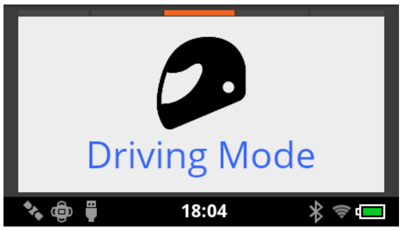

Figure 5.2 – Layout V2 example

Layout 3

Background color:	#3564FC	EXAMPLE
Text color:	#EDEDED	EXAMPLE
Icon color:	#FFFFFF	EXAMPLE

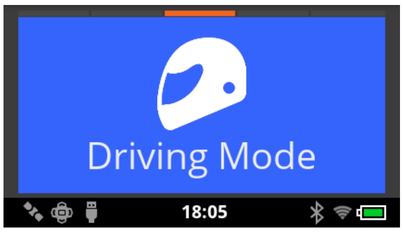

Figure 5.3 – Layout V3 example

All color sets were created to achieve a great level of contrast, and aim to cover most or all of the possible outside environments whilst driving. For example, the first layout would be theoretically more advisable to use in a bright environment, which is consistent with the results obtained in the first experience (see section 7.1.3), and one of the other two layouts would be better suited for nighttime driving.

For navigation indications, one-time events that should draw the drivers' attention, a red background color (EXAMPLE) was chosen, that jointly with a blinking effect results in a screen that is intuitively interpreted as important and urgent.

Finally, notifications have a blue and transparent background color in case of an incoming call (and are superimposed over whatever activity is running), a red and transparent background color in case of a missed or rejected call, and a green and grey color in case of a text message. All notifications' text color is white. Since the utilized call handling system is the one native to the Snow2 system, these colors were not altered, considering they are universally associated to the related events.

5.5 Icons

Icons are an important aspect of the layout, as they are the most easily identifiable component of a whole screen. Thus, they have been carefully selected to instantaneously indicate to the user what information or menu is currently being displayed. Icons are used in the main menu, one for each module, in the navigation inputs and in the weather module. These last ones are default and are provided as a complement to the used weather API. Used icons are the following:

Figure 5.4 – Weather

Figure 5.5 – Tour tracking

Figure 5.6 – Driving mode

Figure 5.7 – Navigation

Figure 5.8 – Call Handling

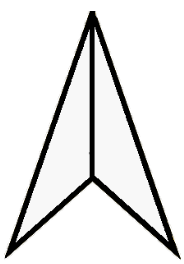

Figure 5.9 - Navigational arrow

Icons are placed in the center of the screen to maximize identification ease, and have a size of 100x100 pixels.

5.6 Menus and tabs

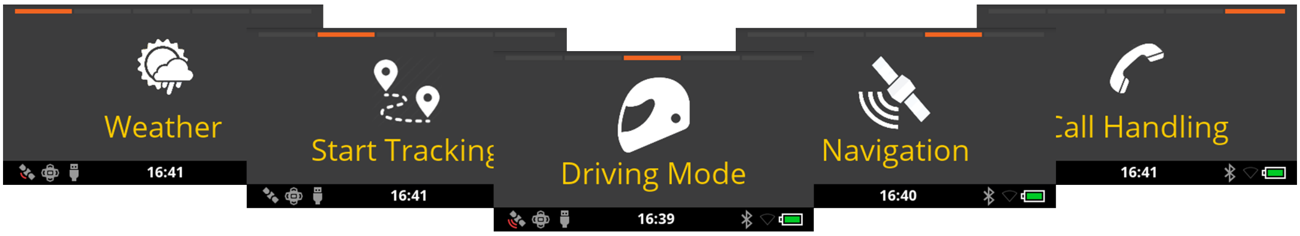

Figure 5.10 – The main menu's various items

The implemented menu system is similar to that of ReconInstruments. Scrolling is horizontal due to the landscape orientation of the screen and a tabbing system exists to give the user awareness of his location in the menu. Tabs are very thin rectangles, placed on top of the screen, and have a gray color if they are not selected, whereas the selected tab is orange, as can be seen in Figure 5.7. In contrast to the native tabbing system, the implemented system does not disappear after a few seconds of idleness, preventing the chance of drivers losing themselves in menus.

As for lists, e.g. the list of destinations or the list of contacts, scrolling is done vertically, and the selected list item is highlighted with a different color. For the contact list specifically, alphabetical scrolling was implemented with left and right input commands, since the list may be of big proportions.

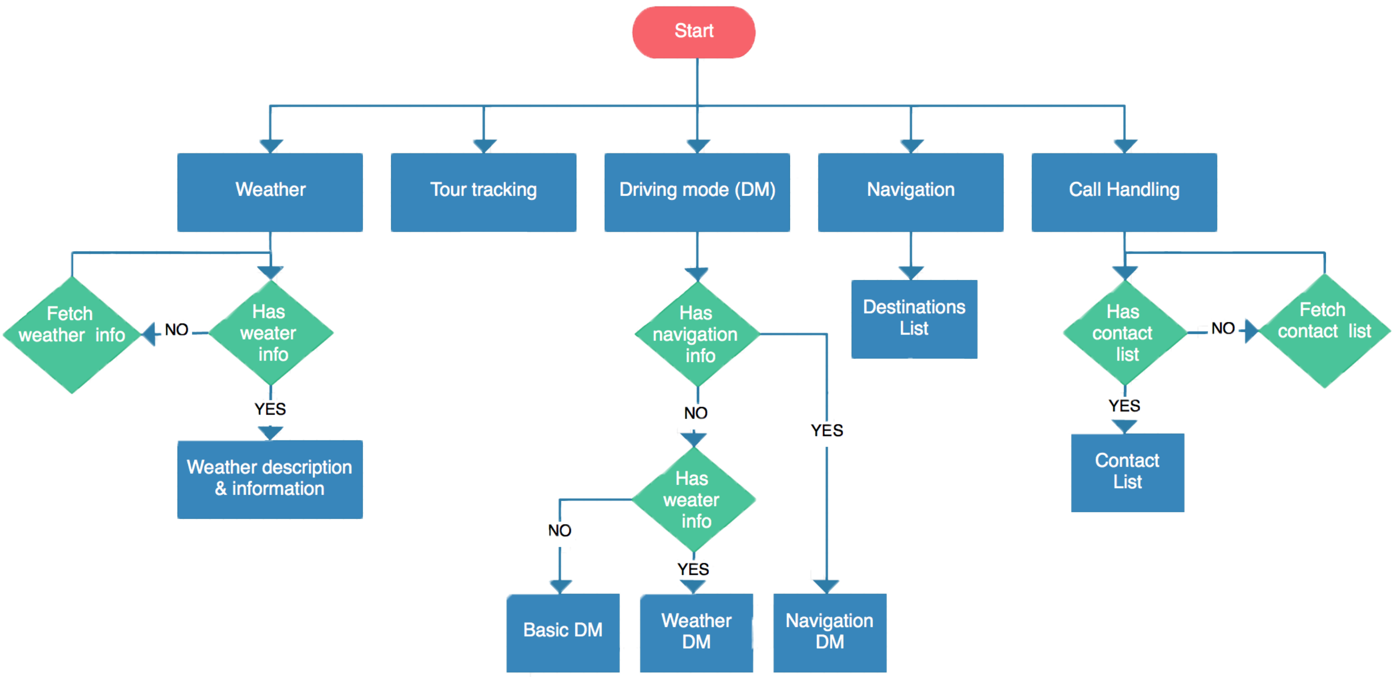

Figure 5.11 – Menu flowchart

5.7 Status bar

A status bar is present throughout the whole application, and constantly displays useful information about general aspects. It is displayed on the bottom of the screen, as can be seen in the various screens of Figure 5.7, and has a black coloured background. At the center, the time is displayed. At the left, a number of icons may appear depending on the settings and status of various parameters. An icon representing the remote control is displayed in white if it is connected, and in red if it is not. If there is GPS connectivity, a directional arrow pointing northeast can be seen, whereas a GPS satellite with waves is shown if there is not. Also, an icon of a smartphone is displayed if a smartphone is connected via Bluetooth to Snow2. Finally, information regarding the battery level is shown on the right side, as well as the Bluetooth icon, which indicates that BT is enabled if the icon is highlighted.

Chapter 6

Methodology

The first step taken towards developing the project was a documentary analysis in the form of a literary review of a large amount of papers that dealt with Head-Up Displays, be it in terms of advantages and disadvantages they represent for users, or in regard to rules and guidelines that should be followed in order to maximize their usefulness and value. This involved filtering documents that in fact were relevant to the project, as well as collecting and putting together all information that was deemed of note as a starting point for the development phase. All relevant data on this subject may be found in Chapter 2.

That data demonstrates that a specific scientific approach to develop a system such as this does not exist, as most studies are independent and no uniformization of information and results exist. Thus, the methodology used in the development of this project was relatively experimental and is a combination of various approaches.

To validate the results obtained thusfar and receive feedback on possible changes or improvements, only one experiment was conducted and a second one has been initially planned and is ready for execution. The experiment comprises a mixture of a questionnaire, observations and an interview. Gathering data through observation alone was not possible, taking into consideration that a) test subjects drove on a motorcycle during tests, precluding the possibility of being in close proximity of the examiner (driving with the examiner on the back of the motorcycle would bring little advantage and was not possible due to technical reasons) and b) application activity could not be visualized due to the nature of the device screen and the device itself. It was still possible and done for a section of the experiment.

Thus, surveys were utilized as a complement to observations to assemble test subjects' experiences and opinions. Since the used technology is not very widespread, a period of adaptation to it was taken into consideration when formulating the questions to be asked, in conjunction with an array of external variables that could potentially affect the outcome of the experiments.

Since questionnaires were answered in the presence of the experiment overseer and all during tests, they would sometimes lead to a form of interview, where one subject would give his opinion on topics that were not in the survey or suggestions for changes and future work. All input, both written and oral, represents an important validation of the project and was taken into great regard. Detailed information on the fulfilled and planned experiments can be found in the next chapter, Chapter 7.

As far as software development is concerned, an incremental build model was used. There is a clear distinction in terms of efficiency between the first and second half of the development phase, as understandably the first weeks were dedicated to adapting to the HUD device and software, the Recon SDK, to researching to what extent various components could be used (the SDK itself, the different sensors), to understanding how exactly the layout should be designed, to developing the first experiment, among other important aspects that seemingly didn't result in any concrete results, but were crucial as an initial planning and research phase. After said phase, modules were woked on iteratively, so that all modules' basic functions were implemented before any further work was completed over existing functionalities. Simultaneous work on more than one module would happen in cases where there were similarities between them, for example the GPS location acquisition for navigation and tour tracking or communication between devices for most modules.

To sum up, the development speed increased exponentially after an initial literary review and adaptation to the development environment, and modules were completed iteratively, either one by one for module-specific functionalities, or at the same time for common requirements. Only one formal evaluation was performed in a form of an experiment.

Chapter 7

Experiments

This chapter documents experiments that were conducted or planned in the scope of the project, including the description of the experiment, its objectives, the satus of the project at the time of the experiment, the results obtained and the conclusions extracted from those results (the last two subjects exist only in case the experiment was conducted). Although they might not have a direct influence on the results, various parameters were recorded and taken into consideration, namely test subject age, driving experience, whether or not gloves were used, if he/she uses glasses or contact lenses and if the motorcycle used has an automatic or manual clutch. Also, answer scales (whenever there was a scale) went from 1, the most negative answer, to 5, the most positive one.

7.1 Intermediate experiment

The intermediate experiment was conducted between 21/04/2015 and 26/04/2015 on a group of 7 male individuals, with an age range between 19 and 49 and and experience range from 1 to 33 years. In all cases but one, there was little to no traffic, and in only one case the trial was performed during night time.

7.1.1 Experiment goals and description

The goals of this experiment consisted of understanding the potential utility of the application for users, and mainly analysing the user experience and interaction with the system. This meant analysing which type of layout drivers preferred, the level of ease in perceiving the information displayed in the HUD and the application's ease of use. Finally, after this short trial of the application in a very primal state, test subjects were asked about possible modifications or improvements to be made in the future.

As for the experiment structure and description, it comprised of three main parts, and a survey was filled out throughout the whole process.

For the first part, drivers did three short predefned laps with the remote control attached to their wrist (they were given the liberty to choose which) and would come back to the starting point after each one to answer the questions related to the completed lap. The tasks to complete were the same for the three laps, the only variable that changed was the layout used. In the first lap, a dark background was used with white icons and orange characters, in the second a white background with black icons and blue characters, and in the third a bright blue background with white icons and characters. In each lap, the test subject was asked to navigate left and right in the main menu, simulate the start or end of a tour (tour tracking at this point was just simulated), and enter Driving mode and observe the displayed instantaneous speed, at the same time comparing it to the speed indicated in the PTW's speedometer.

The second part consisted of a single lap, which was very similar to the ones completed thus far. The diference was simply the location of the remote control, which was placed somewhere on the PTW, mostly on the handles. The objective was to compare the subjects' interaction with the remote control placed on the wrist versus their interaction with it placed on the motorcycle. Originally, the experiment was designed for the drivers to also complete three laps in its second part, but immediately it was concluded that this was not necessary, as the impact on the results would be minimal or probably even inexistent. The first three laps served thus as a comparison for the three different layouts and also as a period of adaption for the subjects to the system, and the second part as a comparison between remote control placement. At the end of the second part's lap, the subjects would again answer to a part of the questionnaire.

Finally, for the third part drivers would complete an unknown route that was indicated through navigation inputs with the HUD. As the navigation module was not yet complete at this point, inputs were given manually by the experiment supervisor, who had to drive behind the test subject and use the remote control to guide his/her way. Correct turns and reaction times were evaluated through the use of the subject's blinkers and naturally the course that was taken. The final part of the survey was then answered.

7.1.2 Project Status

At the time the tests were conducted, the application was still in a fairly rudimentary phase, as far as the implementation of functionalities was concerned. The call handling, weather and tour tracking modules were not yet implemented, and the direction and driving mode modules were still in their early stages. At the time of the experiment's navigation phase, direction indications had to be given manually through the remote control by the person conducting the experiments, since communication between smartphone and HUD was not yet implemented.

Nevertheless, this did not influence the results the tests aimed to collect, since they were more directed towards the drivers' user experience and interaction with the application.

7.1.3 Results

For the first part, there is a clear rise of positive answers as subjects completed more laps, which demonstrates a natural adaptation to the application layout and flow. This is particularly true for the percepted level of difficulty of using the remote control on the wrist. In some cases, there was no improvement until a test subject changed the wrist on which the gear was strapped, which resulted in a positively drastic reaction. One single test subject did not feel comfortable using the remote control on his wrist, possibly due to the fact that he was driving a motorcycle with automatic clutch and only had one year of driving experience, and therefore completed only three laps with the remote control strapped to the motorcycle.

Regarding the layout and how distractive users felt the application was, the tendency was to find it less distractive (generally by 1 point) and find one specific layout better and more perceptive than the other two (also generally by 1 point). The level of distractiveness only worsened if drivers had a very negative reaction to specific layouts, for instance if they found the lighter layouts too bright.

Concluding the first phase of the experiment, the average response level to the question "In this lap, how distractive was the application throughout your driving?" was 3,333 out of 5, to the question "How perceptible is the application with this specific version of the layout?" it was 4,1 out of 5 and to the question "What was the level of difficulty in using the remote control?" it was 3,3 out of 5, which translates into all average answers being above average score.

The second part only consisted of a single lap, so there was no average to calculate, but rather values to compare with the first part. Since drivers could at this point choose the layout they preferred, the answers to the first two questions mentioned above saw an increase of 0.5 and 0.6 respectively, which was to be expected. The most interesting result is the answer to the third question, which increased in a little over 1 whole point. This is a logical result, since the distance from the hand to the remote control is much shorter if it is attached to the motorcycle, and the movement required is also much shorter. Moreover, there is no need to remove the controlling hand from the handle bars, which implies more safety and more practicality. The results seem to confirm these assumptions.

As for the last phase of the trial, two simple questions were asked, considering only the layout aspect of the navigation inputs was being analysed. The first one asked how the subject evaluated GPS navigation inputs and obtained an average score of 3 out of 5. As they answered the survey, some subjects explained they would find these indications more helpful if there was an audio input that accompanied the visual one, and that both together would work best. Also, the suggestion was made to give the inputs not right before a turn or action, but rather with a bigger time interval, which was not done at the time of the experiment. The second question asked how useful/perceptible the GPS navigation inputs were to complete the unknown tour and arrive to the destination. The average result was 4,4 out of 5, which is consistent with the fact that no errors were comitted/ no wrong turns were made, and that the inputs were the only way to find out the correct way. Some drivers pointed to the fact that they had to devote a large amount of attention to the screen in order to avoid making any mistakes. Although this is not ideal, as the driver should pay as much attention to the road ahead as possible (that is exactly one of the main goals of the project), the mentioned additional audio inputs would certainly remedy the situation.

The subject with the most overall negative answers, although possessing decades of experience, did not find his experience with the application a very pleasant one due to a specific set of factors. He was the only one who conducted the experiment at night, which is not very compatible with the orange, darkening lenses of the goggles, that actually had to be taken out for him to feel safe enough to drive witth the goggles. Also, he did not enjoy having the remote on neither wrist (although he found it easy to use on one of them), and the motorcycle mount for the remote control was not very stable and caused the bracelet to budge whenever it had to be used. He was, in fact, the only person who preferred to use the remote on his wrist due to that fact.

In terms of layout, the main aspect that was being examined with these tests, 5 out of 7 drivers preferred the first version of the layout, namely the one with the dark background, white icons and orange characters, and also 5 out of the 7 believed they wouldn't prefer any other version of the layout no matter the weather conditions or luminosity. The layout version preference may be linked with the similar weather and light conditions in which most experiment cases were conducted. When asked about the level of difficulty of reading the application's text/ characters and understanding the icons' meanings, average scores of 4,4 and 4,7 were obtained, respectively.

User experience-wise, also a major factor of analysis, test subjects classified the intuitiveness of the main menu as an average of 4,4 points out of 5, and the intuitiveness of the necessary inputs to navigate through the application (a matter more closely realted to the remote control input types, number of inputs necessary, among other similar aspects) as an average of 4,6 out of 5. All these results indicated a solid and very acceptable form of layout and interaction with the application, with little to no changes to be made in the near future regarding these issues. This does not mean the presented solution is the best possible one or that it is impossible to improve it, but rather that it conferred a good sense of comfort and security to users, which was the main objective. For the system to be launched or realeased to the public, it would probably be advisable to conduct further testing, preferably with use of more and better equipment to monitor subjects' response times, reaction times, and other aspects.

Overall, the results collected from the intermediate experiment were quite positive. Subjects were asked their general reaction to the application at the end, and the response averaged 4 out of 5. Subjects gave an average of 4,1 points out of 5 when asked what the level of difficulty of utilizing the app was (meaning it was quite simple to use). Practically all subjects preferred to have the remote control mounted on the PTW.

7.2 Final experiment

Unfortunately, it was not possible to carry out a second experience in the stipulated time period, even though the system's current state is already stable enough for one. Probably, the conditions under which it would have been carried out would be similar to the previous one, and even the test subject pool would consist of most of the same people.

7.2.1 Experiment goals and description

In contrast with the previous one, this experiment would have functionality evaluation as its main objective, taking into consideration that user experience and user interface have already been validated. With the fact in mind that the project at its current state is a proof of concept, test subjects would probably be asked to use all available features of the system and give their opinion on each module, as well as suggest modifications and improvements. The experiment would probably involve a short drive following navigation inputs to reach a selected destination, a tour marking of (probably) that same drive and its visualization on the smartphone, taking or making a call while driving, asking for weather information, and evaluating the different variations of the driving mode activity. All this process would involve establishing a TCP/IP connection between devices and synchronizing information, the intuitiveness of which would also be tested. Apart from that, it would be ascertained wether or not the various types of inputs (e.g. remote control, navigation through each of the applications) are easy to find and to use. Overall, this experiment would probably have a much larger duration than the first experiment.

7.2.2 Project Status

Project status would be the same as the current project status, which is detailed throughout this whole document, but especially in Chapter 4.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

To summarize, a working prototype of a motorcycle Head-Up Display has been created that performs all core functionalities proposed at the start, and serves as a starting point for future iterations of this work.

Having in mind the nature of the project in development and its possible implications in the users' safety and well being, relevant studies have been analysed and taken into consideration to avoid any accidents and ensure good system usability and experience through a literary review. Aspects that have influence in drivers' perception of and experience with an HUD have been identified and accounted for. Although most literature does not deal directly with PTWs but (mostly) about cars, a strong belief exists that knowledge extracted from it may be applied to PTWs, and in fact it was.

Moreover, existing HUDs for PTWs, both in development and on the market, have been studied and reviewed. This allowed an identification of aspects that are in accordance between all or most of them, which probably indicates their relevance, as well as a combination of features and functionalities that were deemed to be advantageous to the project.

The system is made up of the HUD and the Android application running on it, a Bluetooth remote control that is used to give inputs to said HUD, a smartphone with an auxiliary application and a Bluetooth headset (speakers and microphone). All components are a key part of the solution as a whole, in the sense that some modules would not be functional without one component in the best-case scenario.

It was decided that 5 core modules should make up the system by the end of the project, which were developed interatively. These modules are Weather, Tour Tracking, Driving Mode, Navigation and Call Handling. All requirements necessary to allow the operation of the modules were also implemented, for instance communication between devices, external API calls or GPS location updates. System functionalities generally include weather information about the user's current location, tour tracking and the visualization of the tracked tours, visualization of an array of driving metrics that are useful both during and after driving (for performance analysis), navigation to a selected destination through a set of inputs in real-time, and the possibility to take incoming calls and text messages and make phone calls.

Layout was highly regarded and thought of, so as to find the perfect balance between ease of use, intuitiveness, and the optimal conveyance of information. Due to the size of the screen, only crucial information may be included and its conveyance must be done in a way that is perceived when the driver glances at the screen, because even short periods of staring may be the cause of an accident.

To validate the HUD application and especially its layout component, an experiment was conducted with seven motorcycle drivers using the Snow2 device mounted on the ski goggles. It consisted of several short laps on the street that thus placed test subjects in a real environment, in order to extract valuable and accurate interactions and conclusions. A survey was completed throughout the whole process to collect feedback, after subjects completed different tasks.

Towards the end of of the development process, a prototype was conceived using a modular motorcycle helmet, with Snow2 and a BT headset mounted on it. Even considering the limitations in terms of hardware design and length of cables, the result was positive and the helmet could easily be used in real environment situations.

In conclusion, the expectations and objectives of the project were met to the degree possible. There were no relevant restrictions in terms of software, and all basic functionalities were implemented, though they can and should still be improved (see Section 8.2). The biggest issue is the impossibility to change the hardware design of ReconInstrument's Head-up Display, which has a negative impact on what can be achieved in terms of motorcycle helmet prototype. Affected aspects include how the screen is positioned and consequently how well the driver perceives information displayed in it, and hardware layout within the helmet (length of cables does not allow freedom to place HUD battery wherever necessary, Snow2 device is very bulky and leads to the necessity of modifying/cutting helmet padding). Also, the position of the screen is not ideal and thus it is not a HUD in the real sense that information is superimposed on the user's point of view, but rather the user must very slightly divert his attention every time he wishes to glance at it. Nevertheless, the project as a proof of concept could be considered worthy of investigating and investing by a technology company with the means of improving the referred obstacles. For example, if there was a way to incorporate a small projector in helmets (if it was modular and could attach to any helmet, it would be even better) and display all information on the helmet's visor, not only would results in terms of efficiency, ease of use and safety improve, user acceptance and interest would undoubtedly have a big rise. In this sense, a true HUD would be conceived that could have a big impact on every PTW driver, enhance motorcycling sports and ultimately change the motorcycle driving experience.

However, the achieved results have the same kind of impact, only on a smaller scale. The main goals of ease of use, affordability (this is a cheap solution in comparison to simliar products on the market, and has the advantage of possibly serving other purposes) and utility have been achieved and it is believed that should this system be made available to the general public after its perfection, it would have a generally positive reception.

8.2 Future Work

The system as a whole serves as a proof of concept, but much work is still required for it to be a complete, robust solution that could potentially be made available to the public. In this section, functionalities to be implemented in the near and distant future are presented, divided by modules.

8.2.1 Weather

Currently, weather-related aspects are only partly explored and have the potential to bring much more value to the solution. Mostly, weather information can be an asset in forecasts for future destinations.

In the case that a navigation request is made, an automatic forecast request could be made through openweatherapp's API, in the most basic case for the selected destination, which is fairly useful in itself. Mainly for longer tours, a forecast for intermediate points could also be interesting. This is especially true if there is any case of a storm or bad weather somewhere along the selected route. In this case, an alert could be displayed, warning the driver of the situation beforehand, possibly allowing him/her to change the route or destination, or simply providing the opportunity to wear a raincoat, for example. The issue in terms of implementing such a feature would be to determine how many and which intermediate points to analyse. Even in a single city, there is the possibility that rain falls in some parts and not in others, so a deeper research into geographical meteorology diversification would have to be conducted in order to determine the best way to proceed.

Another related aspect with similar issues would be to periodically evaluate whether there is a case of bad weather in the driver's surroundings. Here, no navigation request would have to be made beforehand, and could be an automatic feature that could be enabled or disabled by the user. Similarly, there could be a check on whether in has rained in the near vicinity. This would prevent drivers from speeding convinced that the pavement is dry, when in fact it may not be. A very specific example for such a case is if it has rained on one side of a hill, and not on the other. If a driver reaches the top of the hill and drives into the wet side of the hill unconscious of the fact, having this feature implemented in the application could avert a fall.

8.2.2 Tour Tracking

The tour tracking module is possibly the one where there is more room to improvement. The current status of this module is at an equal level with others, but there are several useful enhancements that can be made.

The first one has been mentioned in Section 4.2.2, and concerns tour point reduction. As a tour represents tour point numbers that can reach the thousands and even tens of thousands, applying some kind of filter would be very important to maintain an acceptable amount of database entries. The filter that was implemented is an example of how a very simple solution can have a big positive impact on results. Thus, research should be made in order to ascertain what more advances algorithms may be applied to this matter.

At this point, tours can be recorded with the HUD and visualized on the smartphone after a tour synchronization between devices. Nonetheless, the potential utilities of tour tracking go beyond of what has already been accomplished. For instance, tours could be very interesting as shareable content. A simple web server could be developed not only as a way for system users to store their tracked tours, but also to share them with fellow users and search for any available tour that interests them. Even such a popular feature as social sharing could be included. Should the web server be an implementation for the far future, tour sharing between users' smartphone applications can also be a way for friends to share tours. This would be useful e.g. if someone has marked a tour and wants to later retake it with a group of friends. If everyone has the developed solution, the tour could be followed by everyone on their own HUD after the tour has been shared.

Regarding the accuracy of the collected GPS locations that make up a tour, the very recently launched Google Maps Roads API could be a good asset for the system. One of its proposed values is the "Snap to roads" functionality, which takes an array of GPS coordinates and returns a similar array with those points "snapped to the most likely roads the vehicle was traveling along". These points may even be interpolated, resulting in a "path that smoothly follows the geometry of the road" (Google 2015c). We can assume that all marked tours will follow a road, as the system does not support off-track marking. This way, the presented tour will look better and more realistic and could even have an impact on the referred objective of the number of tour point reduction. The only issue found is the fact that this service only accepts a maximum number of 100 points per API call, which implies that several calls would have to be made per tour to smoothen it. Consequences encompass a bigger process time period, but combining the returned data sets and maintaining the tour smoothness wil probably not be affected.

Lastly, tour tracking would be more attractive and complete if the system had an incorporated camera. This would allow pictures to be taken along the tour, either periodically or in strategic points (manually chosen by the driver or somehow selected by the system following a specific algorithm), and associated with the location point of that place. The result would practically be an available preview of tours that would help users evaluate and choose tours in the case that sharing is made available, for instance. The same or possibly even better results could be achieved with short video clips instead of photographs. The inclusion of a camera would be an improvement in itself, and coulb further be used to record or take pictures at any time.

8.2.3 Driving mode (performance monitoring)

Performance monitoring exists in the system only as a limited functionality. For drivers that value driving, fuel consumption and tour optimizations, among other aspects, some improvements to this module could still be accomplished. These improvements have mostly been discussed in Section 4.2.3, and are in respect to calculating fuel consumption by using traveled distance, tank size and number of liters of fuel in a certain formula, or recording maximum and average PTW inclination. Indications could be given to improve driving based on the analysed data, to reduce fuel consumption for example. More specific and detailed data would be seen as valuable by a fraction of all target users, and would confer a more specialized nature to the developed system.

8.2.4 Navigation

Navigation is fundamental and is implemented, but still there is room for overall improvement.

The first aspect to be completed is the implementation of navigation for tours. Tours exist as a collection of points, and requesting directions to follow them is not straightforward. The initial and final tour point cannot be used, as the tour may not be the shortest route between the two. Therefore, some analysis would have to made to solve this issue, possibly having to make several API calls with intermediate tour points. The process of selecting said intermediate points has not yet been thought of, so that its difficulty is still unknown, but hopefully navigation for tours will still be a possibility.

Tour navigation would consist of two parts, namely going from the current location to the beginning of the tour, and the tour itself. The first part corresponds to the navigation that is already available and is thus no issue, but choosing whether to do the tour from beginning to end or from end to beginning should be possible, and that choice will simply affect which point will be selected for the first part and the order of the tour point array. One advantage of the Google Directions API comes into play, which allows requests to contain waypoints. These waypoints are intermediate locations through which the calculated route goes through. Thus, if calculating the strategic waypoints of the tour is possible, the tour route could be maintained while still using the current API and only having to make one request that would only have to be slightly altered. The implemented functions that are responsible for the current navigation module were already thought to support these changes and the inclusion of waypoints.

Another particularity that might be rethought is the current icons that are used for navigation input, very specifically the navigation arrow. As can be seen in Section 5.5, this arrow is very general and only points four ways. Naturally, roads are not that straightforward and in some cases, the current solution might not be enough for the driver to have a very clear idea of what step to take next. In this sense, new arrow designs might be advisable to fill this gap and convey more intuitiveness to this module. One example would be to have the u-turn arrow designed in such a way that a driver knows if the turn should be made by the left or the right side, and another would involve roundabout exits that are commonly many and have very specific layouts.

8.2.5 Miscellaneous

Apart from the already implemented modules, there are a number of features that can be bring extra value to the solution, mainting its original goals and objectives. These features address aspects like imrpoved safety measures, better application performance, leisure, among others.

8.2.5.1 Music player module

One new module that could be added to the project is a music player. With the inclusion of the BT headset and the possibility to play media audio through its speakers, the only work needed would consist of sending information of all music files available on the smartphone to the HUD and listing them. Similarly to the selection of a destination or a contact, the driver could choose a specific music, a message would be sent via TCP/IP to the phone, which would then handle the actual music playback. This list could be reasonably large in terms of size, so that sorting music by artist or albums should also be supported, as well as alphabetical scrolling. This metadata would probably be easily accessible, but could be produced internally otherwise. Other details would also have to be considered, for instance reducing or temporarily pausing the music playback in case a noitification (incoming phone call, text message, navigation input) is received, or if a phone call is made, and resuming it whenever it is deemed appropriate. Regarding the necessary inputs to play, pause, stop, or otherwise control music-related aspects, an evaluation would have to be made and other existing remote control inputs would be factored in in order to maintain a clean, simple and intuitive HCI. The legality of hearing music while driving a PTW was put into question, but since the audio output is through speakers and not earbuds, there is no apparent legal issue, at least in the USA. Nevertheless, it could be fairly distracting to drivers, so a special attention to the planning of this module would be adviseable.

8.2.5.2 Battery life

Thus far, no tests on battery life were performed, neither on the smartphone application nor on the HUD application, mainly because it was not a crucial factor for the scope of this project, and also because the Snow2 device that was used during the development has a worn-out battery, since its only purpose is to serve as a testing and debugging device, thus not producing realistic or reliable results. Notwithstanding, this is an important aspect, especially if the proposed system is brought to market. No user will consider a solution that has a great battery consumption. This is principally true for a product that could be used for long periods of time and cannot be charged during its use, due to practicality and technical reasons.

Thus, battery life optimization should be a priority in a possible late stage of the project development. Having the smartphone application an auxiliary function, but having to be constantly ready for any type of requests, its activity should be kept to a bare minimum during driving periods, since its functionalities will not be required most of the time. Some kind of low energy standby mode should be entered after being idle for a certain period of time, for instance.

With regard to the HUD application, there are various possible ways to minimize application activity and so reduce battery consumption. In principle, GPS location updates are always necessary, so no improvement can be made in that sense. Mostly, battery cost could be cut by reviewing communication with the smartphone, as well as the communication protocol. The number of messages required should be as low as possible; for instance, currently tour synchronization involves sending a single message for each tour point, which can represent thousands of messages. These could possibly be grouped together to drastically reduce message number, which may result in a battery life improvement, not considering the total synchronization period reduction. Apart from this, the overall software might be improved in terms of efficiency, by optimizing loops, altering some system architecture, establishing the necessary device connection only once, among other aspects.

Also regarding the devices' battery, but not its life improvement, handling the case where some device's battery is ending makes sense. Firstly, it would give awareness to the user, for example through notifications containing a warning, both shortly before and right after the fact. This warning would not only mention that a certain device's battery is low, but also serve as a reminder that all functions requiring device communication will not be available as a result. If the HUD runs out of battery, it doesn't even make sense to keep the smartphone running, and otherwise major functions would be unavailable, for instance navigation, new weather information collection or call handling. Secondly, handling this issue would also involve finalizing tasks that are still ongoing, e.g. if a tour is being tracked, it should be finalized before the HUD battery runs out so as not to lose all progress acheived thus far.

8.2.5.3 Automatic settings adjustments

Taking advantage of general information and the available sensors, some conclusions could be determined that would help automatically govern the system, namely adjusting various settings as a benefit for the driver.

Provided the user has a preference for two different layout versions for day time and night time, or a high or low level of brightness, an option could be made available that would automatically change the layout according to those conditions. Data that could support the decision of the layout change are the time in jointly with the date (to take daylight saving time and day length into account), as well as weather information to assess if the sky is overcast.

With reference to audio output in the form of ongoing calls, navigation inputs and possibly music in the near future, it can be a source of distraction for the driver if a crucial task that requires a great amount of focus is returned. Such a task would be, for instance, a tight turn or braking harshly. Should a case like this be identified, sound volume could be reduced, so as to allow the driver to focus on what really is important – driving. After the risk or event is overcome, the volume would be returned to the original level. Motorcycle inclination detected with the gyroscope or a sudden speed reduction could be interpreted as signs to execute the volume reduction.

8.2.5.4 Security measures

Security is one of the key points of the project at hand. Drivers' safety should not only be preserved, but preferably also improved. With this in mind, a simple and yet potentially life-saving feature would be an emergency contact in case of a crash detection.

An accident may easily be deduced from a very sudden speed reduction or strange levels of inclination and rotation, and PTW drivers run the risk of a serious injury, which is much higher than that of car drivers (see section 2.1). In critical situations such as these, a difference in seconds could mean the difference between life or death, in a worst-case scenario. A proposed functionality would deduce the crashes in the referred way and issue a check on whether the driver is hurt and needs medical attention in the form of a pop-up. Should there be no answer in a fixed period of time, an automatic emergency call would be made to the national emergency number, or alternatively a text message would be sent either to that same emergency number or a predefined emergency contact. Whatever the form of the contact, the voice or text message would contain all the relevant information for an ambulance to be called. Previously, some of this information would have had to be recorded in the application by user input, like name, age, allergies or blood type, and other could be determined by the system, specifically the user's location that is crucial for the task at hand. The conditions under which this emergency contact is made must be undoubtedly confirmed and assessed as true, since the allocation of a medical team to someone that does not need its help represents a serious matter.

Again assuming the possibility of having a camera included in the system, it could be used to record accidents/crashes and later serve as proof of what happened. The camera could start recording as soon as a the crash detection mechanism gave a signal, and stop and store the results after a specific period of time, to allow the capture of the whole event.

8.2.5.5 Helmet prototype

The presented helmet prototype serves as a proof of concept, but is not yet close to a possible final product. Therefore, some improvements on it may be made, as has been referred in Section 3.3.5.1. There is not much room to improve if ReconInstrument's device maintains its exact structure, as it barely leaves a margin to change the current prototype display. A partnership with ReconInstruments to come up with a more PTW-dedicated device would have the most impact on the obtained results, though this expectation is not realistic.

Presently, there are no prototypes for an open helmet and an integral helmet, though the latter's structure would be very silimar to the developed proof of concept.

The probable best way to validate the current prototype is to conduct an experiment and survey a pool of users on its usefulness, practicality and comfort.

References

Ablassmeier, M., T. Poitschke, F. Wallhoff, K. Bengler, and G. Rigoll. 2007. "Eye Gaze Studies Comparing Head-Up and Head-Down Displays in Vehicles." Multimedia and Expo, 2007 IEEE International Conference on, 2-5 July 2007.

Alves, P. R. J. A., J. Goncalves, R. J. F. Rossetti, E. C. Oliveira, and C. Olaverri-Monreal. 2013. "Forward collision warning systems using heads-up displays: Testing usability of two new metaphors." Intelligent Vehicles Symposium Workshops (IV Workshops), 2013 IEEE, 23-23 June 2013.

BIKEHUD. 2014. "The world's only augmented reality Head-Up Display for your motorcycle helmet." Accessed February 2015. http://www.bike-hud.com.

Changrak, Yoon, Kim Kyongho, Park Hye Sun, Park Min Woo, and Jung Soon Ki. 2014. "Development of augmented forward collision warning system for Head-Up Display." Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, 8-11 Oct. 2014.

Doshi, A., Cheng Shinko Yuanhsien, and M. M. Trivedi. 2009. "A Novel Active Heads-Up Display for Driver Assistance." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 39 (1):85-93. doi: 10.1109/TSMCB.2008.923527.

Gale, A.G., I.D. Brown, S.P. Taylor, and C.M. Haslegrave. 1998. "Defining the hud benefit time window." In *Vision in Vehicles VI*. Amsterdam: Elsevier.

Google. 2015a. "Google Maps Android API." Accessed 2015. https://developers.google.com/maps/documentation/android/.

Google. 2015b. "Google Maps Directions API." Accessed 2015. https://developers.google.com/maps/documentation/directions/.

Google. 2015c. "Google Maps Roads API." Accessed 2015. https://developers.google.com/maps/documentation/roads/.

Horrey, W. J. , C.D. Wickens, and A.L. Alexander. 2003. "The Effects of Head-Up Display Clutter and In-Vehicle Display Separation on Concurrent Driving Performance." Proceedings of the Human Factors and Ergonomics Society Annual Meeting, October.

Jun, Ma, Zhou Hang, and Xu Wenxia. 2014. "Study on driving simulation based ADAS design." Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2014 IEEE Conference and Expo, Aug. 31 2014-Sept. 3 2014.

Mobile, Vision. 2014. "European App Economy 2014." Accessed February. http://www.visionmobile.com/product/european-app-economy-2014/.
NUVIZ. 2015a. "THE FIRST HEAD-UP DISPLAY FOR YOUR MOTORCYCLE HELMET." Accessed February 2015. http://www.ridenuviz.com/ - intro.
NUVIZ. 2015b. "Frequently asked questions." Accessed February 2015. http://www.ridenuviz.com/faqs/.
OpenWeatherMap. 2015. "Weather API." Accessed 2015. http://openweathermap.org/api.
Pieve, M., F. Tesauri, and A. Spadoni. 2009. "Mitigation accident risk in powered two Wheelers domain: Improving effectiveness of human machine interface collision avoidance system in two wheelers." Human System Interactions, 2009. HSI '09. 2nd Conference on, 21-23 May 2009.
Realm. 2015. "Realm is a mobile database." Accessed 2015. https://realm.io.
Schweigert, M. 2004. "The eye-tracking system JANUS." Ph.D., Institute of Ergonomics, Munich University of Technology.
SKULLY. 2015a. "Get the gear." Accessed February 2015. http://store.skully.com.
SKULLY. 2015b. "The world's smartest motorcycle helmet." Accessed February 2015. http://www.skully.com/ - smartest-helmet.
Todoriki, T., J. Fukano, S. Okabayashi, M. Sakata, and H. Tsuda. 1994. "Application of head-up displays for in-vehicle navigation/route guidance." Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994, 31 Aug-2 Sep 1994.
Ward, N. J., A. M. Parkes, and P. R. Crone. 1994. "The effect of background scene complexity on the legibility of head-up-displays for automotive applications." Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994, 31 Aug-2 Sep 1994.

Anexo A

Survey results

Below is a table with all asked questions and each of the test subjects' answers. For confidentiality purposes, their names have been replaces by letters. Answers are either Yes/No, or from a range of 1-5, if not a direct answer to the question (e.g. Years of experience). In answers that range between 1-5, 1 is the most negative answer and 5 the most positive one.

<table>
<tr><th></th><th colspan="3">A</th><th colspan="3">B</th><th colspan="3">C</th><th colspan="3">D</th><th colspan="3">E</th><th colspan="3">F</th><th colspan="3">G</th></tr>
<tr><td colspan="22">Before trials</td></tr>
<tr><td>Age</td><td colspan="3">23</td><td colspan="3">22</td><td colspan="3">19</td><td colspan="3">49</td><td colspan="3">36</td><td colspan="3">22</td><td colspan="3">23</td></tr>
<tr><td>Years of experience</td><td colspan="3">2</td><td colspan="3">1</td><td colspan="3">3</td><td colspan="3">33</td><td colspan="3">20</td><td colspan="3">5</td><td colspan="3">7</td></tr>
<tr><td>Do you use glasses or contact lenses or have any vision deficiency?</td><td colspan="3">No</td><td colspan="3">Yes</td><td colspan="3">No</td><td colspan="3">Yes</td><td colspan="3">No</td><td colspan="3">Yes</td><td colspan="3">No</td></tr>
<tr><td>Chosen wrist for remote control</td><td colspan="3">Left</td><td colspan="3">Left</td><td colspan="3">Right</td><td colspan="3">Left</td><td colspan="3">Left</td><td colspan="3">Right</td><td colspan="3">Left</td></tr>
<tr><td>Will you use gloves?</td><td colspan="3">No</td><td colspan="3">Yes</td><td colspan="3">No</td><td colspan="3">Yes</td><td colspan="3">No</td><td colspan="3">No</td><td colspan="3">No</td></tr>
<tr><td>Type of motorcycle gears</td><td colspan="3">Auto</td><td colspan="3">Manual</td><td colspan="3">Auto</td><td colspan="3">Auto</td><td colspan="3">Auto</td><td colspan="3">Manual</td><td colspan="3">Auto</td></tr>
<tr><td colspan="22">First phase (3 laps, each w/ a different layout version, remote on the wrist)</td></tr>
<tr><td rowspan="2">During this lap, how distractive was the application while driving?</td><td>1</td><td>2</td><td>3</td><td>1</td><td>2</td><td>3</td><td>1</td><td>2</td><td>3</td><td>1</td><td>2</td><td>3</td><td>1</td><td>2</td><td>3</td><td>1</td><td>2</td><td>3</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>2</td><td>2</td><td colspan="3">-</td><td>4</td><td>5</td><td>5</td><td>3</td><td>2</td><td>2</td><td>1</td><td>5</td><td>5</td><td>3</td><td>3</td><td>3</td><td>3</td><td>4</td><td>4</td></tr>
</table>

<table>
<tr><td>How perceptible is the application with this version of the layout?</td><td>4</td><td>5</td><td>4</td><td colspan="3">-</td><td>5</td><td>5</td><td>5</td><td>5</td><td>3</td><td>4</td><td>1</td><td>5</td><td>5</td><td>3</td><td>3</td><td>3</td><td>5</td><td>4</td><td>5</td></tr>
<tr><td>What is the level of difficulty of using the remote control?</td><td>2</td><td>3</td><td>3</td><td colspan="3">-</td><td>3</td><td>4</td><td>4</td><td>3</td><td>2</td><td>2</td><td>1</td><td>5</td><td>5</td><td>5</td><td>5</td><td>3</td><td>2</td><td>2</td><td>5</td></tr>
<tr><td colspan="22">First phase (1 lap, remote mounted on the motorcycle)</td></tr>
<tr><td rowspan="2">During this lap, how distractive was the application while driving?</td><td colspan="3" rowspan="2">5</td><td>1</td><td>2</td><td>3</td><td colspan="3" rowspan="2">5</td><td colspan="3" rowspan="2">2</td><td colspan="3" rowspan="2">4</td><td colspan="3" rowspan="2">3</td><td colspan="3" rowspan="2">3</td></tr>
<tr><td>5</td><td>5</td><td>5</td></tr>
<tr><td>How perceptible is the application with this version of the layout?</td><td colspan="3">4</td><td>4</td><td>5</td><td>4</td><td colspan="3">5</td><td colspan="3">5</td><td colspan="3">5</td><td colspan="3">5</td><td colspan="3">5</td></tr>
<tr><td>What is the level of difficulty of using the remote control?</td><td colspan="3">5</td><td>1</td><td>3</td><td>3</td><td colspan="3">5</td><td colspan="3">5</td><td colspan="3">3</td><td colspan="3">5</td><td colspan="3">5</td></tr>
<tr><td colspan="22">Second phase (1 lap w/ navigation inputs)</td></tr>
<tr><td>How do you evaluate GPS indications?</td><td colspan="3">3</td><td colspan="3">2</td><td colspan="3">3</td><td colspan="3">4</td><td colspan="3">3</td><td colspan="3">3</td><td colspan="3">3</td></tr>
<tr><td>How perceptible/ useful were GPS indications to</td><td colspan="3">4</td><td colspan="3">4</td><td colspan="3">5</td><td colspan="3">4</td><td colspan="3">5</td><td colspan="3">4</td><td colspan="3">5</td></tr>
</table>

arrive to your destination?							
After trials							
General reaction to the application	3	5	4	5	3	4	4
What's the level of difficulty of using the application?	4	3	4	4	4	5	5
How slow was traffic?	4	4	5	1	5	5	5
How bright was it outside?	5	4	5	3	2	4	3
What were the weather conditions?	Sunny	Sunny	Sunny	Overcast	Night	Overcast	Rain
Considering your last two answers, which layout version did you prefer?	1	2	1	1	3	1	1
Do you think you would prefer another layout version under other conditions?	No	No	No	Yes	Yes	No	No
How easily could you read the application's texts?	5	5	5	4	3	4	5
How easily did you understand the icons'	5	5	5	4	4	5	5

meanings?							
How intuitive is the main menu?	4	5	4	5	5	4	4
How intuitive are the necessary inputs to navigate through the application?	5	5	5	5	2	5	5
In what position did you prefer to use the remote control?	Handle-bars	Handle-bars	Handle-bars	Handle-bars	Wrist	Handle-bars	Handle-bars
How precise is the application, comparing the instantaneous speed shown by it with the one shown on the motorcycle's speedometer?	4	4	4	5	2	4	3